SPEECH AND LANGUAGE TECHNOLOGIES

Edited by **Ivo Ipšić**

INTECHWEB.ORG

Speech and Language Technologies
Edited by Ivo Ipšić

Published by InTech
Janeza Trdine 9, 51000 Rijeka, Croatia

Statements and opinions expressed in the chapters are these of the individual contributors and not necessarily those of the editors or publisher. No responsibility is accepted for the accuracy of information contained in the published articles. The publisher assumes no responsibility for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained in the book.

Publishing Process Manager Iva Lipovic
Technical Editor Teodora Smiljanic
Cover Designer Jan Hyrat
Image Copyright Marko Poplasen, 2010. Used under license from Shutterstock.com

First published June, 2011
Printed in India

A free online edition of this book is available at www.intechopen.com
Additional hard copies can be obtained from orders@intechweb.org

Speech and Language Technologies, Edited by Ivo Ipšić
p. cm.
ISBN 978-953-307-322-4

Contents

Preface

The book “Speech and Language Technologies” addresses state-of-the-art systems and achievements in various topics in the research field of speech and language technologies. Book chapters are organized in different sections covering diverse problems, which have to be solved in speech recognition and language understanding systems.

In the first section machine translation systems based on large parallel corpora using rule-based and statistical-based translation methods are presented. The third chapter presents work on real time two way speech-to-speech translation systems.

In the second section two papers explore the use of speech technologies in language learning.

The third section presents a work on language modeling used for speech recognition.

The chapters in section Text-to-speech systems and emotional speech describe corpus-based speech synthesis and highlight the importance of speech prosody in speech recognition.

In the fifth section the problem of speaker diarization is addressed.

The last section presents various topics in speech technology applications, like audio-visual speech recognition and lip reading systems.

I would like to thank to all authors who have contributed research and application papers from the field of speech and language technologies.

Ivo Ipšić
University of Rijeka
Croatia

Part 1

Machine Translation

Towards Efficient Translation Memory Search Based on Multiple Sentence Signatures

Juan M. Huerta
IBM TJ Watson Research Center,
United States

1. Introduction

The goal of machine translation is to translate a sentence *S* originally generated in a source language into a sentence *T* in target language. Traditionally in machine translation (and in particular in Statistical Machine Translation) large parallel corpora are gathered and used to create inventories of sub-sentenial units and these, in turn, are combined to create the hypothesis sentence *T* in the target language that has the maximum likelihood given S. This approach is very flexible as it has the advantage generating reasonable hypotheses even when the input has not resemblance with the training data. However, the most significant disadvantage of Machine Translation is the risk of generating sentences with unnaceptable linguistic (i.e., syntactic, grammatical or pragmatic) incosistences and imprefections.

Because of this potential problem and because of the availability of large parallel corpora, MT researchers have recently begun to expore the *direct* search approach using these translation databases in support of Machine Translation. In these approaches, the underlying assumption is that if an input sentence (which we call a *query*) *S* is sufficiently similar to a previously hand translated sentence in the memory, it is, in general, preferable to use such existing translations over the generated Machine Translation hypothesis. For this approach to be practical there needs to exist a sufficiently large database, and it should be possible to identify and retrieve this translation in in a span of time comparable to what it takes for the Machine Translation engine to carry out its task. Hence, the need of algorithms to efficiently search these large databases.

In this work we focus on a novel approach to Machine Translation memory lookup based on the efficient and incremental computation of the string edit distance. The string edit distance (SED) between two strings is defined as the number of operations (i.e., insertions, deletions and substitutions) that need to be applied on one string in order to transform it into the second one (Wagner & Fischer, 1974). The SED is a symmetric operation.

To be more precise, our approach leverages the rapid elimination of unpromising candidates using increasingly stringent elimination criteria. Our approach guarantees an optimal answer as long as this answer has an SED from the query smaller than a user defined threshold. In the next section we first introduct string similarity translation memory search, specifically based on the string edit distance computation, and following we present our approach which focuses on speeding up the translation memory search using increasingly stringent sentence signatures. We then describe how to implement our approach using a Map/Reduce framework and we conclude with experiments that illustrate the advantages of our method.

2. Translation memory search based on string similarity

A translation memory consists of a large database of pre-translated sentence pairs. Because these translation memories are typically collections of high quality hand-translations developed for corpus building purposes (or in other cases, created for the internationalization of documentation or for other similar development purpoes), if a sentence to be translated is found *exactly* in a translation memory, or within a relativelly small edit distance from the query, this translation is preferred over the output generated by a Machine Translation engine. In general, because of the nature of the errors introduced by SMT a Translation Memory match with an edit distance smaller than the equivalent average BLEU score of a SMT hypothesis is preferred. To better understand the relationship between equivalend BLEU and SED the reader can refer to (Lin & Och, 2004).

Thus, for a translation memory to be useful in a particular practical domain or application, 3 conditions need to be satisfied:

- It is necessary that human translations are of at least the same average quality (or better) than the equivalemtn machine translation output
- It is necessary that there is at least some overlap between translation memory and the query sentences and
- It is necessary that the translation memory search process is not much more computationally expensive than machine translation

The first assumption is typically true for the state of the current technology and certainly the case when the translations are performed professional translators. The second condition depends not only on the semantic overlap between the memory and the queries but also on other factors such as the sentence length distribution: longer sentences have higher chances of producing matches with larger string edit distances nullifying their usefulness. Also, this condition is more common in certain domains (for example technical document translation where common technical processes lead to the repeated use of similar sentences). The third assumption, (searching in a database of tens of millions of sentences) is not computationally trivial especially since the response time of a typical current server-based Machine Translation engine is of about a few hundred words per second. This third requirement is the main focus of this work.

We are not only interested in finding exact matches but also in finding *high-similarity* matches. Thus, the trivial approach to tackle this problem is to compute the string edit distance between the sentence to be translated (the source sentence) and *all* of the sentences in the database. It is easy to see how when comparing 2 sentences each with length *n* and using Dynamic Programming based String Edit Distance (Navarro, 2001) the number of operations required is $O(n^2)$. Hence, to find the best match in a memory with *m* senteces the complexity of this approach is $O(mn^2)$. It is easy to see how a domain where a database contains tens of millions of translated sentences and where the average string length is about 10, the naive search approach will need to perform in the order of billions of operations *per query*. Clearly, this naive approach is computationally innefficient.

Approximate string search can be carried out more efficiently. There is a large body of work on efficient *approximate* string matching techniques. In (Navarro, 2001), there is a very extensive overview of the area of approximate string matching approaches. Essentially, there are two types of approximate string similarity search algorithms: on-line and off-line. In the on-line methods, no preprocessing is allowed (i.e., no index of the database is built). In off-line methods an index is allowed to be built prior to search.

We now provide, as background, an overview of existing methods for approximate string match as well as advantages and disadvantages of these in order to position our approach in this context.

2.1 Off-line string similarity search: index based techniques

Off line string similarity approaches typically have the advantage of creating an index of the corpus prior to search (see for example (Bertino et al., 1997). These approaches can search, in this way, for matches much faster. In these methods terms can be weighted by their individual capability to contribute to the overall recall of documents (or, in this case, sentences) such as TD-IDF or Okapi BM25.

However, index-based approaches are typically based on so called bag-of-words distance computation in which the sentences are converted into vectors representing only word count values. Mainly because of their inability to model word position information, these approaches can only provide a result without any guarantee of optimality. In other words, the best match returned by an index query might not contain the best scoring sentence in the database given the query. Among the reasons that contribute to this behavior are: the non-linear weights of the terms (e.g., TF-IDF), the possible omission of stop words, and primarily the out-of-order nature of the bag of words approach.

To overcome this problem, approaches based on positional indexes have been proposed (Manning, 2008). While these indexes are better able to render the correct answer, they do so at the expense of a much larger index and a considerable increase in computational complexity. The complexity of a search using a positional index is *O(T)* where T denotes the number of tokens in the memory *(T=nm)*. Other methods combine various index types like positional, next word and bi-word indices (e.g., (Williams et al., 2004)). Again, in these cases accuracy is attained at the expense of computational complexity.

2.2 On-line string similarity matching: string edit distance techniques

There are multiple approaches to on-line approximate string matching. These, as we said, do not benefit from an index built *a-priori*. Some of these approaches are intended to search for exact matches only (and not approximate matches). Examples of on line methods include methods based on Tries (Wang et al., 2010) (Navarro & Baeza-Yates, 2001), finite state machines, etc. For an excellent overview in the subject see (Navarro, 2001).

Our particular problem, translation memory search, requires the computation of the string edit distance between a candidate sentence (a query) and a large collection of sentences (the translation memory). Because as we saw in the previous section the string edit distance is an operation that is generally expensive to compute with large databases, there exist alternatives to the quick computation of the string edit distance. In order to better explain our approach we first start by describing the basic dynamic programming approach to string edit distance computation.

Consider the problem of computing the string edit distance between two strings *A* and *B*. Let *A={a_1,...a_n}* and *B={b_1,...b_m}*. The dynamic programming algorithm, as explained in (Needleham & Wunsch, 1970), consists of computing a matrix *D* of dimensions *(m+1)x(n+1)* called the edit-distance-matrix, where the entry *D[i,j]* is the edit distance *SED(Ai,Bj)* between the prefixes A_i and B_j. The fundamental dynamic programming recurrence is thus,

$$D[i,j] = \min\left\{\begin{array}{cc} D[i-1,j]+1 & \text{if } i>0 \\ D[i,j-1]+1 & \text{if } j>0 \\ D[i-1,j-1]+\partial_{a,b} & \text{if } i>0 j>0 \end{array}\right\} \quad (1.1)$$

The initial condition is *D[0,0]=0*. The edit distance between *A* and *B* is found in the lower right cell in the matrix *D[m,n]*. We can see that the computation of the Dynamic Programming can be carried out in practice by filling out the columns (j) of the DP array. Figure 1 below, shows the DP matrix between sentences Sentence$_1$="A B C A A" and Sentence$_2$="D C C A C" (for simplicity words are considered in this example to be letters). We can see that the distance between these two sentences is 3, and the cells in bold are the cells that constitute the optimal alignment path.

2.2.1 Sentence pair improvements on methods based on the DP matrix

A taxonomy of approximate string matching algorithms based on the dynamic programming matrix is provided in (Navarro, 2001). Out of this taxonomy, two approaches are particularly relevant to our work. The first is the approach of Landau Vishkin 89, which focusing on a Diagonal Transition manages to reduce the computation time to *O(kn)* where *k* is the maximum number of expected errors *(k<n)*. The second is Ukkonen 85b which based on a cutoff heuristic also reduces the computation time to *O(kn)*. Our work is based on a multi-signature approach that uses ideas similar to the heuristics of Ukkonen.

	i	0	1	2	3	4	5
i			**A**	**B**	**C**	**A**	**A**
0		**0**	1	2	3	4	5
1	**D**	1	**1**	2	3	4	5
2	**C**	2	2	**2**	2	4	5
3	**C**	3	3	3	**2**	3	4
4	**A**	4	3	4	3	**2**	3
5	**C**	5	4	4	4	3	**3**

Fig. 1. Sample Dynamic Programming Matrix

2.2.2 Approximate string edit distance computation

In section 2.1 we described an off-line method for segment retrieval based on an index and a bag of words approach. That approach does not intend to approximate the string edit distance; rather, it calculates similarity based on term distribution vectors and thus produces results that differ from the SED approach. To reduce this mismatch between off-line and on-line methods, it is possible to approximate the SED (and the related Longest Common Subsequence computation) based on a stack computation and information derived from a positional index. This computation is possible through the use of a Stack structure and a A* like algorithm as described in (Huerta, 2010b). In that paper, Huerta proposed a method that takes *O(m s log s)* operations on average where s is the depth of the stack (typically much smaller than *T*, or *m*) instead of *O(T)* using a positional index. This approach is important because it improves the accuracy of an off line system using a position index by using an approximation of the string edit distance without sacrificing speed. The results are within 2.5% error (Huerta, 2010b). In this paper, we will focus exclusively on the on-line approach.

3. Multi-signature SED computation

Our approach is intended to produce the best possible results (i.e., find the best match in a translation memory given a query if this exists within a certain *k*, or number of edits) at speeds comparable to those produced by less accurate approaches (i.e., indexing), in a way that is efficiently parallelizable (specifically, implementable in Map Reduce). To achieve this, our approach decomposes the typical single DP computation into multiple consecutive string signature based computations in which candidate sentences are rapidly eliminated. Each signature computation is much faster than any of its subsequent computations.

The core idea of the signature approach is to define a hypersphere of radius equal to *k* in which to carry out the search. In other words, a cutoff is used to discard hypotheses. Eventually the hypersphere can be empty (without hypotheses) if there is no single match within the cutoff (whose distance is smaller than the cutoff).

The idea is that, at each signature stage the algorithm should be able to decide efficiently with certainty if a sentence lies outside the hypersphere. By starting each stage with a very large number of candidates and eliminating a subset, the algorithm shows the equivalent of perfect recall but its precision only increases with a rate inversely proportional to the running speed. The signature based algorithms (the kernels) are designed to be very fast at detecting out of bound sentences and slower at carrying out exact score computations. We start by describing the 3 signature based computations of our approach.

3.1 Signature 1: string length

The first signature computation is carried out taking into account the length of the query string as well as the length of the candidate string. This first step is designed to eliminate a large percentage of the possible candidates in the memory very rapidly. The idea is very simple: a pair of strings S1 and S2 cannot be within *k* edits if |*l1*-*l2*|>k, where *l1* is the length of string 1 and so on.

Figure 1 below shows a histogram of the distribution of the length of a translation memory consisting of 9.89 Million sentence pairs. As we can see, the peak is at sentences of length 9 and consists of 442914 sentences which correspond to about 4.5% of the sentences. But the average length is 14.5 with a standard deviation of 9.18, meaning that there is a long tail in the distribution (a few very long sentences). We assume that the distribution of the query strings matches that of the memory. The *worst case*, for the particular case of *k*=2, constitutes the bins in the range *l1-k*<*l2*<*l1*+*l2*. This in our case is the case of the query equals to 9. For this particular case and memory the search space is reduced to 2.18M (i.e., to 22% and hence reduced by 78%). This is in the worst case that happens only 4.5% of the time. The weighted average improvement is a reduction of the search space to 10% of the original. This, in turn speeds up search on average 10x.

An even faster elimination of candidates is possible if multiple values of *k* are used depending on the length of the memory hypotheses. For example one can run with a standard k for hypotheses larger than 10 and a smaller k for hypotheses smaller or equal to 10. One can see from the distribution of the data that the overall result of this first signature step is the elimination of between 70% to more than 90% of the possible candidates, depending on the specific memory distribution.

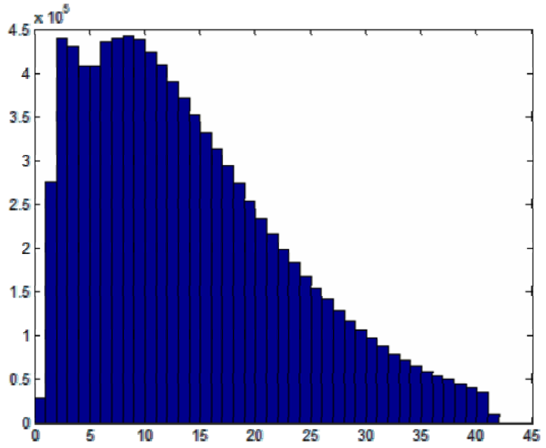


Fig. 2. Histogram of distribution of sentence lengths for a Translation Memory

3.2 Signature 2: lexical distribution signature

The second signature operation is related to the lexical frequency distribution and consists of a fast match computation based on the particular words of the query and the candidate sentences. We leverage the Zipf-like distribution (Ha et al., 2001) of the occurrence frequency of words in the memory and the query to speed up this comparison. To carry out this operation we compute the sentence lexical signature, which for Sentence *Si* is a vector of length *li* consisting of the words in the sentence sorted by decreasing rarity (i.e., increasing frequency). We describe in this section an approach to detect up to k-differences in a pair of sentence signatures in time (worst case) less than *O(n)* (where n is the sentence average length). The algorithm (shown in figure 3 below) stops as soon as *k* differences between the signature of the query and the signature of the hypothesis are observed, and the particular hypothesis is eliminated.

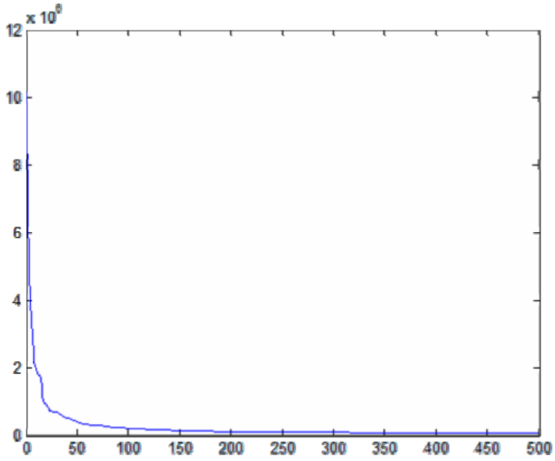


Fig. 3. Frequency distribution of words in the sample Translation Memory

To better motivate our algorithm let us explain a little bit the distribution of the words in the translation memory we use as an example. First, we address the question, how unique is the lexical signature of a sentence? Empirically, we can see in our sample translation memory that out of the 9.89M sentences, there are 9.85M unique lexical signatures, meaning that this information by itself could constitute a good match indicator. Less than 1.0% of the sentences in the corpus share a lexical signature with another sentence. As we said, the signature is based on the Zipf's distribution of word frequencies. It has been observed (Ha et al) that at least for the case of the most frequent words in a language is inversely proportional to rank.

We now describe the algorithm to efficiently compute the bound in differences in lexical items between two sentences.

```
Search Algorithm
construct the sorted vector of instances for A and B. Sa and Sb
i=0,j=0, d=0;
while number of differences is less than k  (d<k)
   if Sa[i]==Sb[j]
            then
             i++; j++; break;
    else if Sa[i]>Sb[j]
           d++;
           j++;
   else
         d++;
        i++;
end while
```

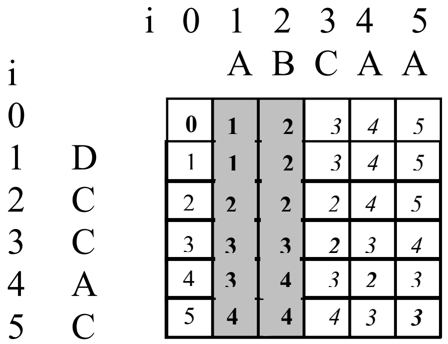
Fig. 4. Search Algorithm for Lexical Frequency Distribution String Signature

It is possible to show that the above code will stop (quit the *while* loop) on an average proportional to *O(alpha k)* where *alpha* is bounded by characteristics of the word frequency distribution. Also, the worst case is *O(n)* (when the source is equal to the target) this will happen with an empiric probability of less than 0.01 in a corpus like the one we use. The best case is *O(k)*.

As we will see in later sections, this signature approach combined with Map-Reduce produces very efficient implementations: the final kernel performs an exact computation passing from the Map to the Reduce steps only those hypotheses within the radius. The Reduce step finds the best match for every given query by computing the DP matrix for each query/hypothesis pair. The speedup in this approach is proportional to the ratio of the volume enclosed in the sphere divided by the whole volume of the search space.

3.3 Signature 3: bounded dynamic programming on filtered text

After the first two signature steps, a significant number of candidate hypotheses have been eliminated based on string length and lexical content. In the third step a modified Dynamic Programming computation is performed over all the surviving hypotheses and the query. The matrix based DP computation has two simple differences from the basic algorithm described before. The first one instructs the algorithm to stop after the minimum distance in an alignment is *k* (i.e., when the smallest value in last column is k) and sets its focus on the diagonal. The second modification relates to the interchange of the sentences so that the longest sentence in the columns and the shortest one in the rows. Figure 5 below shows an example of a DP matrix in which in the second column is obvious that the minimum alignment distance is at least 2. If, for this particular case, we were interested in k<2, then the algorithm would need to stop at this point. An additional difference is also possible and further increases the speed: each sentence in the memory (and the query itself) are represented by non-negative integers, where each integer represents a word id based on a dictionary. In our experiments we used very large dictionaries (400k), in which elements such as URL's and other special named entities are all mapped to the *unknown word* ID.



		i 0	1 A	2 B	3 C	4 A	5 A
i							
0		0	1	2	3	4	5
1	D	1	1	2	3	4	5
2	C	2	2	2	2	4	5
3	C	3	3	3	2	3	4
4	A	4	3	4	3	2	3
5	C	5	4	4	4	3	3

1

Fig. 5. Bounded DP Matrix

3.4 Combining signatures: representation of the memory

We have described how to carry out the multi-signature algorithm. While this approach significantly increases the search speed, for it to be truly efficient in practice, it should avoid computing the signature information of the translation memory for each query it receives. Rather it should use a slightly bigger pre-computed data structure in which for each sentence, the length signature and the lexical signature are available.

The translation memory will thus consist of one record for each sentence in the memory. Each record in the memory will consist of the following fields: The first field has the sentence length. The second field has the lexical signature vector for a sentence. The third field has the dictionary filtered memory sentence. The fourth field has the plain text sentence. While this representation increases the size of the memory by a factor of at least 3, we have found that it is extremely useful in keeping the efficiency of the algorithm.

4. Map/Reduce parallelization

We previously described that our multi-signature algorithm can be further sped-up by carrying it out in a parallelized fashion. In this section we describe how to do so based on the Map/Reduce formulation, specifically on Hadoop.

Map-Reduce (and in particular Hadoop) (Dean & Ghemawat, 2008) is a very useful framework to support distributed in large computer clusters. The basic idea is to segment the large dataset (in our case, the memory) and provide portions of this partition to each of the worker nodes in the computing cluster. The worker nodes perform some operation on the segment of the partition domain and provides results in a data structure (a hash map) consisting of key-value pairs. All the output produced by the worker nodes is collated and post-processed in the Reduce step, where a final result set is obtained. An illustration of the general process is shown in figure 6.

In our particular Map Reduce implementation the translation memory constitutes the input records that are partitioned and delivered to the map worker nodes. Each Map job reads the file with the translation queries and associates each with a key. Using the multi signature approach described above, each worker node rapidly evaluates the SED-feasibility of candidate memory entries and, for those whose score lies within a certain cutoff, it creates an entry in the hash map. This entry has as the key the query sentence id, and as the value a structure with the SED score and memory id entry.

In the reduce step, for every query sentence, all the entries whose key correspond to the particular query sentence in question are collated and the best candidate (or top N-best) are selected. Possibly, this set can be empty for a particular sentence if no hypotheses existed in the memory within k-edits.

It is easy to see that if the memory has *m* records and the query set has *q* queries the maximum set of map records is *qm*. Hadoop sorts and collates these map records prior to the reduce step. Thus in a job where the memory has 10M records and the query set ha 10k sentences, the number of records to sort and to collate are 100Billion. This is a significantly large collection of data to organize and sort. It is crucial to reduce this number and as we cannot reduce *q* our only alternative is to reduce *m*. That is precisely the motivation behind the multi-signature approach. The multi-signature approach that is proposed in this work not only avoids the creation of such large number of Map records but also reduces the exact Dynamic Programming computation spent in the Map jobs.

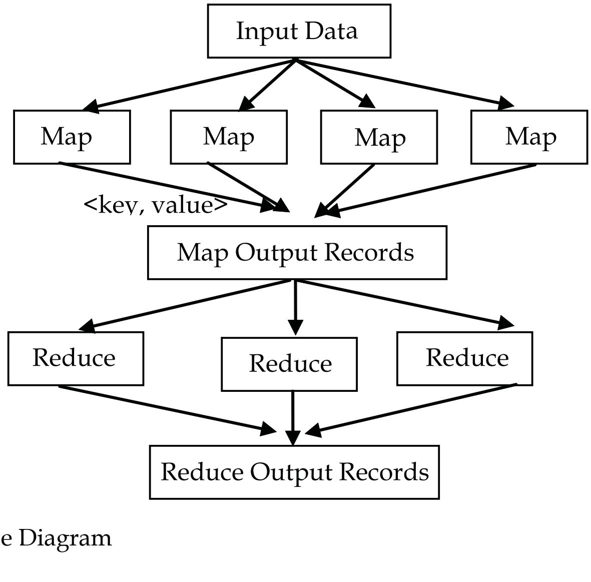


Fig. 6. Map Reduce Diagram

5. Experiments

To conclude, this section binds all the previous themes by providing a set of experiments describing the exact string similarity computation speed and coverage results. To test our algorithms we explored the use of an English-to-Spanish Translation memory consisting of over 10M translation pairs.

Our test query sentences consist of a set of 7795 sentences, comprising 125k tokens (words). That means the average query length is 16.0 words per sentence. Figure 7 shows the histogram of the distribution of the lengths in the query set. We can see in this histogram that there are 2 modes: one is for sentences of length 3 and the other is for sentences of length 20.

Our Hadoop environment for Map Reduce parallelization consists of 5 servers: 1 name node and 4 dedicated to data node servers (processing). The name nodes have 4 cores per CPU resulting in a total of 16 processing cores. We partitioned the memories in 16 parts and carried out 16 map tasks, 16 combine tasks (which consolidate the output of the map step prior to the reduce task) and 1 reduce task. The file system was an HDFS (Hadoop Distributed File System) consisting on one Hard Drive per server (for a total of 5 Hard Drives).

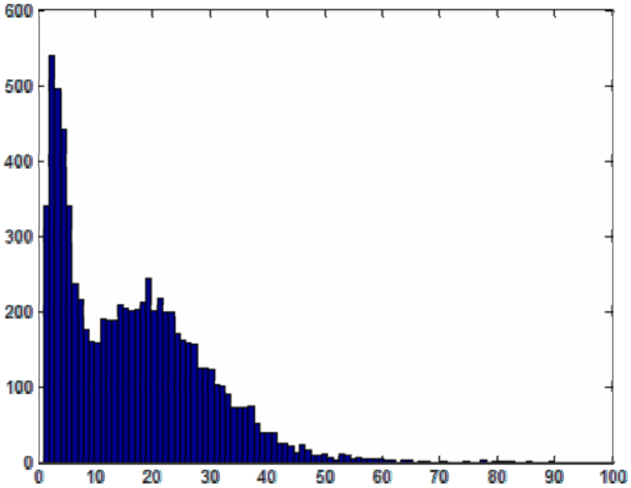


Fig. 7. Histogram of distribution of sentence lengths for a Query Set

We ran two sets of experiments using various cutoff configurations. In the first set of configurations, the map-reduce jobs use a single cutoff that is applied equally to all the query sentences and hypotheses. In the second configuration for each sentence apply one of 2 cutoffs depending on the length of each query.

Table 1 shows the results for the case of the single cutoff (Cutoff 1= Cutoff 2). Column 1 and 2 correspond to the cutoff (which is the same), in Column 3 we can see the number of queries found, Column 4 shows the total time it took to complete the batch of queries (in seconds), and Column 5 shows the total number of Map records. As we can see the as we increase the cutoff the number of output sentences, the time and the number of map records increase. We will discuss in more detail the relationship between these columns. Table 2 shows the same results but in this case Cutoff 1 (for short queries) is not necessarily equal to Cutoff 2 (for long queries).

Cutoff 1	Cutoff 2	Output Sentences	Time (s)	Map Records
1	1	773	148	773
2	2	1727	170	10.24M
3	3	2355	411	277M
4	4	2749	898	1.04B
5	5	3056	1305	2.048B
6	6	3285	2145	3.14B

Table 1. Experimental results for Cutoff1=Cutoff2

Cutoff 1	Cutoff 2	Output Sentences	Time (s)	Map Records
2	5	2770	271	20.4M
2	6	2999	475	344M
2	7	3271	581	716M

Table 2. Experimental results for length specific Cutoff

We can see that if we are allowed to have two length-related cutoffs, the resulting number of output sentences is kept high at a much faster response time (measured in seconds per query). So for example if one wanted to obtain 2700 sentences we can use cutoff 2 and 5 and run in 271 seconds or alternatively use 4 and 4 and run in 898 seconds. The typical configuration attains 2770 sentences (cutoffs 2 and 5) in 271 seconds for an input of size 7795 which means 34 ms per query. This response time is typical, or faster that a translation engine and this allows for our approach to be a feasible runtime technique.

Figure 8 shows the plot of the total processing time for the whole input set (7795 sentences) as a function of the input cutoff. Interestingly, one can see that the curve follows a non-linear trend as a function of the cutoff. This means that as the cutoff increases, the number of operations carried out by our algorithm increases non-linearly. But, how exactly are these related?

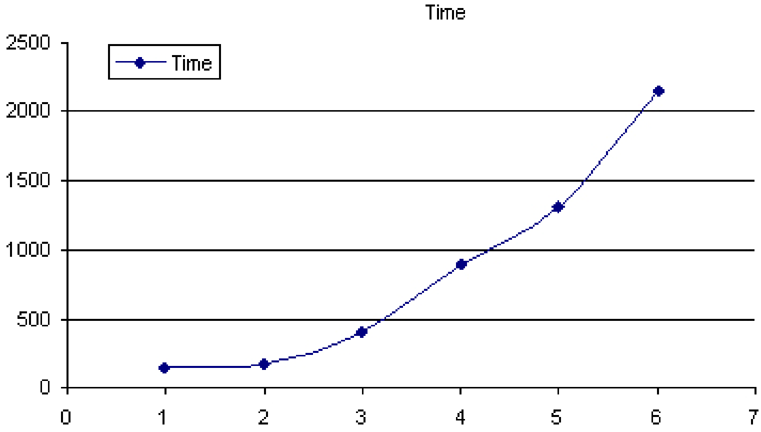


Fig. 8. Time as a function of single cutoff

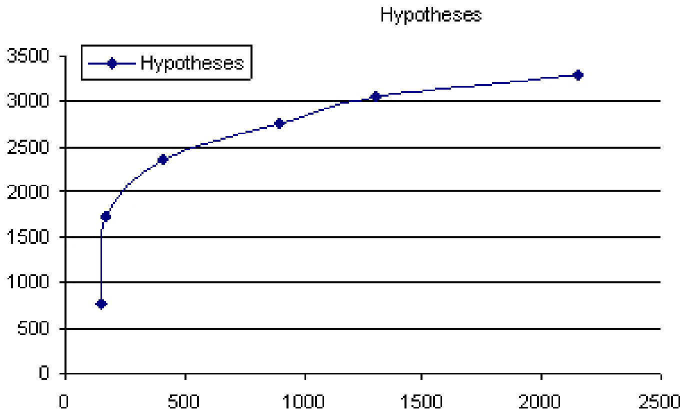


Fig. 9. Number of Output Hypotheses as a Function of Total Run Time

To explore this question, Figure 9 shows the total number of output sentences as a function of total runtime. We see that in the first points there is a large increase in output hypotheses per additional unit of time. After the knee in the curve, though, it seems that the system stagnates as it reduces its output per increment in processing time. This indicates that the growth in processing time that we are observing by increasing the threshold is not the direct result of more output hypotheses being generated. As we will se below, rather it is the result of a growth in processing map records.

In Figure 10 we show the total number of map records (in thousands) as a function of observed total run-time. Interestingly, this is an almost linear function (the linear trend is also shown).As we have mentioned, the goal of our algorithm is to minimize the records it produces. Having minimized the number of records, we have effectively reduced the run-time.

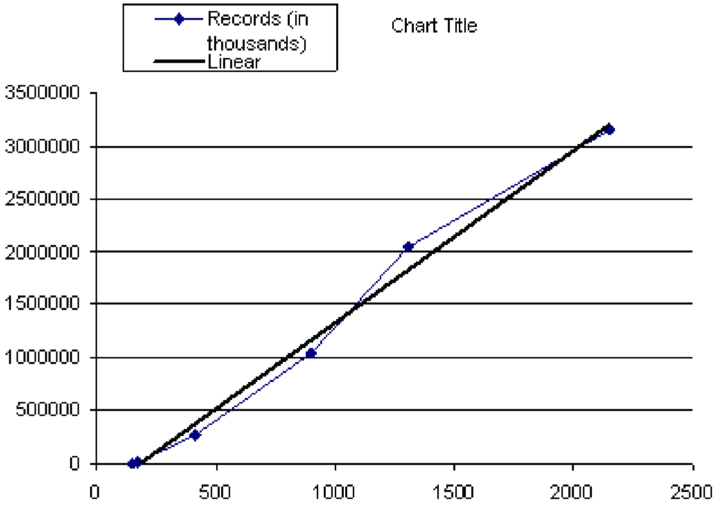


Fig. 10. Number of Map Records (in thousands) as a Function of Total Run Time

Finally Figure 11 shows the number of records per number of output hypotheses. This figure tells us about the efficiency of the system in producing more output. We can see that as the system strives to produce more output matches a substantially larger number of records needs to be considered considerably increasing the computation time.

6. Conclusion

We have presented here an approach to translation memory retrieval based on the efficient search in a translation pair database. This approach leverages several characteristics of natural sentences like the sentence length distribution, the skewed distribution of the word occurrence frequencies as well as DP matrix computation optimization into consecutive sentence signature operations. The use of these signatures allows for a great increase in the efficiency of the search by removing unlikely sentences from the candidate pool. We demonstrated how our approach combines very well with the map reduce approach.

In our results we found how the increase in running time experienced by our algorithm as the cutoff is increased grows in a non-linear way. We saw how this run time is actually related to the total number of records handled by Map Reduce. Therefore it is important to reduce the number of unnecessary records without increasing the time to carry out the signature computation. This is precisely the motivation behind the multi-signature approach described in this work.

The approach described in this paper can be applied directly into other sentence similarity approaches, such as sentence-based Language Modelling based on IR (Huerta, 2011) and others where large textual collections are the norm (like Social Network data, (Huerta, 2010)). Finally, this approach can also be advantageously extended to other non-textual domains in which the problem consists of finding the most similar sequence (e.g., DNA sequences etc) where the symbol frequency distributions of the domain sequences is skewed and there is a relatively broad sequence length distribution.

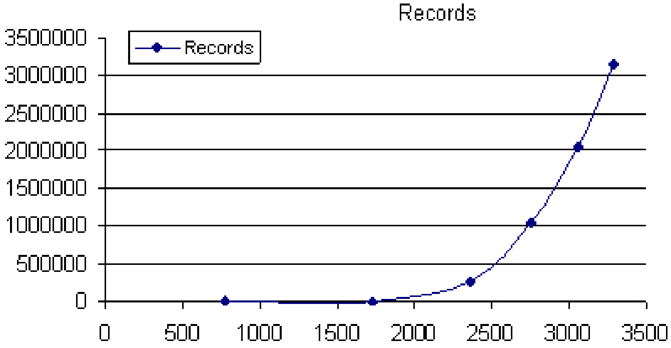


Fig. 11. Number of Map Records as a Function of Total Output Hypotheses

7. References

Bertino E., Tan K.L., Ooi B.C., Sacks-Davis R., Zobel J., & Shidlovsky B. (1997). *Indexing Techniques for Advanced Database Systems*. Kluwer Academic Publishers, Norwell, MA, USA

Dean J., & Ghemawat S. (2008). *MapReduce: simplified data processing on large clusters*. Commun. ACM 51, 1 (January 2008), 107-113

Ha L. Q., Sicilia-Garcia E. I., Ming J., & Smith F. J. (2002). *Extension of Zipf's law to words and phrases*. In Proceedings of the 19th international conference on Computational linguistics - Volume 1 (COLING '02), Vol. 1. Association for Computational Linguistics, Stroudsburg, PA, USA, 1-6.

Huerta J. M. (2011), *Subsequence Similarity Language Models*, Proc. International Conference in Acoustics Speech and Signal Processing 2011

Huerta J. M. (2010), *An Information-Retrieval Approach to Language Modeling: Applications to Social Data* NAACL #Social Media Workshop

Huerta J. M. (2010b) *A Stack Decoder Approach to Approximate String Matching* In Proc. SIGIR 2010

Landau G. M., Myers E. W., & Schmidt J. P. (1998). Incremental String Comparison. SIAM J. Comput. 27, 2 (April 1998), 557-582.

Lin C. W., & Och F. J. (2004). *Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics*. In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics ACL '04

Manning C. D., Raghavan P., & Schütze H. (2008). *Introduction to Information Retrieval*, Cambridge University Press. 2008.

Navarro G., Baeza-Yates R., Sutinen E., & Tarhio J. (2001). *Indexing methods for approximate string matching,* IEEE Data Engineering Bulletin, 2001

Navarro G. (2001). *A guided tour to approximate string matching*, ACM Computing Surveys v.33 No. 1 2001

Needleham S.B. & Wunsch C.D. (1970). *A general method applicable to the search for similarities in the amino acid sequence of two proteins*, J. of Mol. Bio. Vol 48 (1970), pp. 443-453.

Wagner R.A. & Fischer M. J. (1974). *The string-to-string correction problem,* J. of the ACM, vol. 21 No. 1 (1974) pp-168-173

Wang, J. Fang J. & Li G. (2010). *TrieJoin:Efficient Triebased String Similarity Joins with EditDistance Constraints* The 36th International Conference on Very Large Data Bases, September 1317,

Williams H.E., Zobel J., & Bahle D. (2004). *Fast Phrase Querying with Combined Indexes*. ACM Transactions on Information Systems, 22(4) , 573-594, 2004

2

Sentence Alignment by Means of Cross-Language Information Retrieval

Marta R. Costa-jussà[1] and Rafael E. Banchs[2]

[1] *Barcelona Media Innovation Center, Spain*
[2] *Institute for Infocomm Research, Singapore*

1. Introduction

In this chapter, we focus on the specific problem of sentence alignment given two comparable corpora. This task is essential to some specific applications such as parallel corpora compilation Utiyama & Tanimura (2007) and cross-language plagiarism detection Potthast et al. (2009).

We address this problem by means of a cross-language information retrieval (CLIR) system. CLIR deals with the problem of finding relevant documents in a language different from the one used in the query. Different strategies are used, from ontology based Soerfel (2002) to statistical tools. Latent Semantic Analysis can be used to get a list of parallel words Codina et al. (2008). Multidimensional Scaling projections Banchs & Costa-jussà (2009) can also be used in order to find similar documents in a cross-lingual environment. Other techniques are based on machine translation, where the search is performed over translated texts Kishida (2005). Within this framework, two basic components should be distinguished: a translation model, and a retrieval model that may work as in the monolingual case. The translation can be faced either in the query, or in the document. In the case of document translation, statistical machine translation systems can be used for translating document collections into the original query language. In the case of query translation, the challenges of deciding how a term might be written in another language, which of the possible translations should be retained, and how to weight the importance of translation alternatives when more than one translation is retained should be considered.

Here, we use the query translation approach. Then, a segment of text in a given source language is used as query for recovering a similar or equivalent segment of text in a different target language. Given that we are using complete sentences which provide a certain context for the terms to be translated, we do not have the disadvantages mentioned in the above lines. Particularly, when using the query translation approach, we investigate if using either a rule-based or a statitical-based machine translation system influence the final quality of the sentence alignment. Additionally, we test if standard automatic MT metrics are correlated with the standards metrics of the sentence alignment.

Rule-based machine translation (RBMT) systems were the first commercial machine translation systems. Much more complex than translating word to word, these systems develop linguistic rules that allow the words to be put in different places, to have different meaning depending on context, etc. RBMT technology applies a set of linguistic rules in three

different phases: analysis, transfer and generation. Therefore, a rule-based system requires: syntax analysis, semantic analysis, syntax generation and semantic generation.
Statistical Machine Translation (SMT), a corpus-based approach, is a more complicated form of word translation, where statistical weights are used to decide the most likely translation of a word. Modern SMT systems are phrase-based rather than word-based, and assemble translations using the overlap in phrases.

2. Organization of the chapter

The rest of this chapter is structured as follows. Next section describes several sentence alignment approaches. Section 4 reports the motivation of our CLIR approach. Section 5 describes in detail how our sentence alignment system works. Section 6 describes the two machine translation approaches that are used and compared in this chapter: rule-based and statistically-based. Next, experimental framework and the proposed methodology are illustrated by performing cross-language text matching at the sentence level on a tetra-lingual document collection. Also, within this section, the performance quality of the implemented systems is compared, showing that in this application the statistical system provides better results than the rule-based system. Section 8 reports the translation quality of both translation systems and reports the correlation among translation quality and cross-language sentence matching quality. Finally, in section 9, most relevant conclusions derived from the experimental results are presented.

3. Related work

Sentence alignment has been approached from different perspectives. In the following subsections we briefly describe some well-known methods.

- Gale & Church (1993) proposed a sentence alginer provided a probability score for each sentence pair based on sentence-length (number of characters). Their method use dynamic programming to find maximum likelihood alignment.
- The Bilingual Sentence Aligner Moore (2002) combines sentence length based method with word correspondence. It makes a first pass based on sentence length and a second pass based on IBM Model-1. The former is based on the distribution of length variable and the latter is trained during runtime and uses alignments obtained from the first pass. The larger corpus size, the more effective (better model of distribution of word length variable and word correspondence).
- Hunalign Varga et al. (2005) uses the diagonal of the alignment matrix, plus a bias of 10%. The weights are a combination of length-based and dictionary-based similarity. If there is no dictionary, they do length-based, estimate dictionary from result and reiterate once. The main problems is that it is not designed to handle corpora of over 20k sentences, it copes by splitting larger corpora and this causes worse dictionary estimates.
- Gargantua Braune & Fraser (2010) is an alignment model similar to Moore (2002), but it introduces differences in pruning and search strategy.
- Bleualign Senrich & Volk (2010) is based on automatic translation of source text. It uses dynamic programming to find path that maximizes BLEU scorePapineni et al. (2001) between target text and translation of source text.

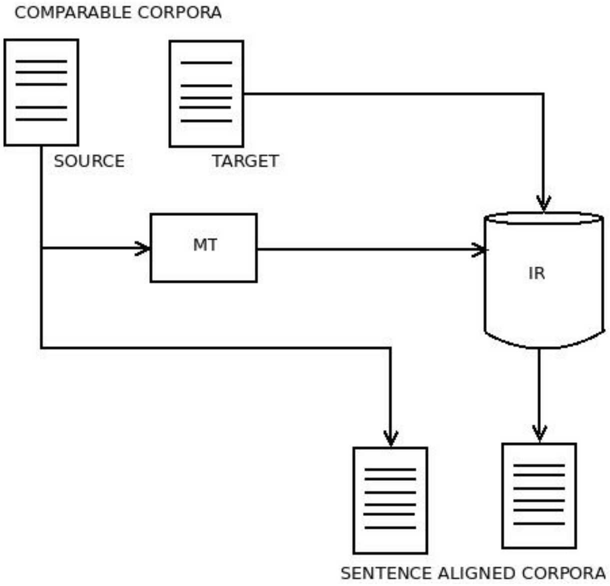


Fig. 1. Block Diagram of the CLIR approach for Sentence Alignment.

4. Motivation

CLIR systems are becoming more and more accurate due to the improvement in machine translation and information retrieval quality. As fas as we are concerned, CLIR have never been used before for sentence alignment. However, with this study, we are demonstrating that it is a nice shot to try. Building a CLIR system is relatively easy if using available tools.

In addition to testing a new methodology for sentence alignment, we want to experiment with different machine translation systems. Particularly, we want to compare two translation systems from different core technologies: rule-based and statistical. This two types of MT commit different types of errors, which may have different effects on the sentence alignment challenge. Although it is not objective of this work, we also report the correlation between translation quality in terms of BLEU and sentence alignment quality.

5. Sentence alignment based on cross-language information retrieval

A cross-language information retrieval (CLIR) system can be used for sentence alignment. The idea is to use a sentence as a query and search for the indexed sentence that matches best. One of the most popular systems in CLIR is the query translation approach which consists of concatenating a machine translation system and a monolingual information retrieval system. See the block diagram in Figure 1.

Basically, an information retrieval (IR) system uses a query to find objects that are indexed in a database. Several documents may match the same query but with different degrees of relevance. In order to make information retrieval efficient, the queries and documents are typically transformed into a suitable representation. One of the most popular representations is the vector space model where documents and queries are represented as vectors, each

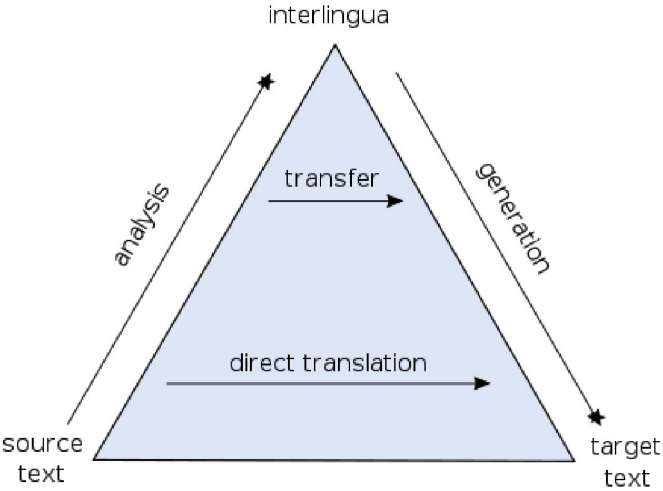


Fig. 2. Machine translation approaches.

dimension corresponding to a separate term. Usually, terms are weighted with the term frequency and inverse document frecuency (tf-idf) scheme.

The main challenge in CLIR with respect to IR is that the query language is different from the document language. We approach the problem of sentence aligning by operating a machine-translation-based CLIR system at the sentence level over a bilingual comparable corpus. In this context, we are comparing the performance of two machine translation systems with different core technologies: rule-based and statistical.

6. Machine translation core technologies

As mentioned, there are different core tecnologies in machine translation. Corpus-based approaches (such as Statistical) use a direct translation and rule-based approaches use a transfer translation. See Figure 2 [1]. As follows we briefly describe the two technologies.

6.1 Rule-based machine translation

Rule-based machine translation (RBMT) systems develop linguistic rules that allow the words to be put in different places, to have different meaning depending on context, etc. The Georgetown-IBM experiment in 1954 was one of the first rule-based machine translation systems and Systran was one of the first companies to develop RBMT systems.

RBMT methodology applies a set of linguistic rules in three different phases: analysis, transfer and generation. Therefore, a rule-based system requires: syntax analysis, semantic analysis, syntax generation and semantic generation. In general terms, RBMT generates the target text given a source text following the next steps.

Given a source text, the first step is to segment it, for instance, by expanding elisions or marking set phrases. These segments are then looked up in a dictionary. This search returns

[1] http://en.wikipedia.org/wiki/Machine_translation

the base form and tags for all matches (morphological analyser). Afterwards, the task is to resolve ambiguous segments, i.e. source terms that have more than one match, by choosing only one (part of speech tagger). Additionally, a RBMT system may add a lexical selection to choose between alternative meanings. After the module taking care of the lexical selection, two modules follow, namely the structural and the lexical transfers. The former consists of looking up disambiguated source-language base work to find the target-language equivalent. The latter consists in: (1) flagging grammatical divergences between source language and target language, e.g. gender or number agreement; (2) creating a sequence of chunks; (3) reordering or modifying chunk sequences; and (4) substituting fully-tagged target-language forms into the chunks. Then, tags are used to deliver the correct target language surface form (morphological generator). Finally, the last step is to make any necessary orthographic change (post-generator).

One of the main problems of translation is choosing the correct meaning, which involves a classification or disambiguation problem. In order to improve the accuracy, it is possible to apply a method to disambiguate meanings of a single word. Machine learning techniques automatically extract the context features that are useful for disambiguating a word.

RBMT systems have a big drawback: the construction of such systems demands a great amount of time and linguistic resources, thus resulting very expensive. Moreover, in order to improve the quality of a RBMT it is necessary to modify rules, which requires more linguistic knowledge. The modification of one rule cannot guarantee that the overall accuracy will be better. However, using rule-based methodology may be the only way to build an MT system when dealing with minor languages, given that SMT requires massive amounts of sentence-aligned parallel text. RBMT may use linguistic data without access to existing machine-readable resources. Moreover, it is more transparent: errors are easier to diagnose and debug.

6.2 Statistical machine translation

Statistical Machine Translation (SMT), which started with the CANDIDE system Berger et al. (1994), is, at its most basic, a more complicated form of word translation, where statistical weights are used to decide the most likely translation of a word. Modern SMT systems are phrase-based rather than word-based, and assemble translations using the overlap in phrases. The main goal of SMT is the translation of a text given in some source language into a target language by maximizing the conditional propability of the translated sentence given the source one. A source string $s_1^J = s_1 \ldots s_j \ldots s_J$ is translated into a target string $t_1^I = t_1 \ldots t_i \ldots t_I$. Among all possible target strings, the goal is to choose the string with the highest probability:

$$\tilde{t_1^I} = \underset{t_1^I}{argmax}\, P(t_1^I | s_1^J)$$

where I and J are the number of words in the target and source sentences, respectively.

The first SMT systems were reformulated using Bayes' rule. In recent systems, such an approach has been expanded to a more general maximum entropy approach in which a log-linear combination of multiple feature functions is implemented (Och, 2003). This approach leads to maximising a linear combination of feature functions:

$$\tilde{t} = \underset{t}{argmax} \left\{ \sum_{m=1}^{M} \lambda_m h_m(t,s) \right\}.$$

The job of the translation model, given a target sentence and a foreign sentence, is to assign a probability that t_1^I generates s_1^J. While these probabilities can be estimated by thinking about how each individual word is translated, modern statistical MT is based on the intuition that a better way to compute these probabilities is by considering the behavior of phrases (sequences of words). The phrase-based statistical MT uses phrases as well as single words as the fundamental units of translation. Phrases are extracted from multiple segmentations of the aligned bilingual corpora and their probabilities are estimated by using relative frequencies.

The translation problem has also been approached from the finite-state perspective as the most natural way for integrating speech recognition and machine translation into a speech-to-speech translation system (Bangalore & Riccardi, 2000; Casacuberta, 2001; Vidal, 1997). The Ngram-based system implements a translation model based on this finite-state perspective (de Gispert & Mariño, 2002) which is used along with a log-linear combination of additional feature functions (Mariño et al., 2006).

In addition to the translation model, SMT systems use the language model, which is usually formulated as a probability distribution over strings that attempts to reflect how likely a string occurs inside a language (Chen & Goodman, 1998). Statistical MT systems make use of the same *n*-gram language models as speech recognition and other applications do. The language model component is monolingual, so acquiring training data is relatively easy.

The lexical models allow the SMT systems to compute another probability to the translation units based on the probability of translating word per word. The probability estimated by lexical models tends to be in some situations less sparse than the probability given directly by the translation model. Many additional feature functions can also be introduced in the SMT framework to improve the translation, like the word or the phrase bonus.

6.3 Challenges of RBMT and SMT

State-of-the-art rule-based MT approaches have the following challenges:

- *Semantic*. RBMT approaches concentrate on a local translation. Usually, this translation tends to be literal and it lacks of fluency. Additionally, words may have different meanings depending on their grammatical and semantic references.
- *Lexical*. Words which are not included in the dictionary will have no translation. When keeping the system updated, new language words have to be introduced in the dictionary.

State-of-the-art statistical MT approaches have the following challenges:

- *Syntactic*. The main challenge in this category is word reordering, which can be of two natures: long reordering, as when translating between languages with different structures (SVO versus VSO), and short reorderings, as such involving relative locations of modifiers and nouns Costa-jussà & Fonollosa (2009); Tillmann & Ney (2003); Zhang et al. (2007).
- *Morphological*. Here there are chanllenges as gender and number agreement. For instance, keeping number agreement when translating from English to Spanish in structures such as*Noun + Adjective* de Gispert et al. (2006); Nießen & Ney (2004).
- *Lexical*. Here there are the Out-of-Vocabulary words which can not be translated. The main causes of out of vocabulary words is the dependency with the training data. In most SMT approaches, the limitation of training data, domain changes and morphology are not taken into account. Approaches such as the one from Langlais & Patry (2007) try to deal with these challenges.

The semantic and lexical problems may affect more to a CLIR system than the syntactic and morphological errors, taking into account that IT systems work with bag-of-words and use words and stems.

7. Experiments

As already mentioned in the introduction, in this work, we focus on the problem of sentence alignment given two comparable corpora. In this particular task, a segment of text in a given source language is used as query for recovering an equivalent segment of text in a different target language. In this section, we evaluate a conventional query translation approach first described by Chen & Bao (2009) which considers a cascade combination of a machine translation system and a monolingual IR system. We use two machine translation systems with different core technologies: a rule-based and a statistical-based machine translation systems.

7.1 Multilingual sentence dataset

The dataset considered for the experiments is a multilingual sentence collection that was extracted from the Spanish Constitution, which is available for downloading at the Spanish government's main web portal: *www.la-moncloa.es*. In this website, all constitutional texts are available in five different languages, including the four official languages of Spain: Spanish, Catalan, Galego and Euskera, as well as English. Given that the MT systems used do not provide Euskera translation, we limited the experiments to four languages. The texts are organized in 169 articles plus some additional regulatory dispositions. All texts were segmented into sentences and the resulting collection was filtered according to sentence length. More specifically, sentences having less than five words were discarded aiming at eliminating titles and some other non-relevant information. Moreover, we had to perform a manual postprocessing to correct some errors in the sentence alignment. Table 1 summarizes the main statistics for both the overall collection.Table 2 shows a sentence example.

Collection	English	Spanish	Catalan	Gallego
Sentences	611	611	611	611
Running words	15285	14807	15423	13760
Vocabulary	2080	2516	2523	2667
Average sent. length	25.01	24.23	25.24	22.52

Table 1. Corpus statistics.

7.2 Evaluation of the methodology

The system to be considered implements a query translation strategy followed by a standard monolingual information retrieval approach.
For the query translation step, we used the following MT systems:

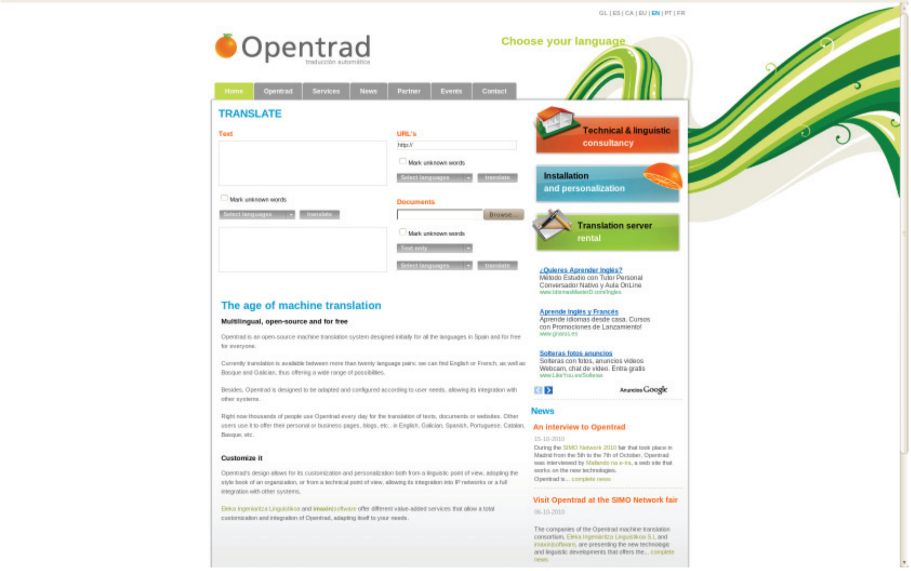
1. A rule-based system implemented with the Opentrad platform[2]. This system Ramírez-Sánchez et al. (2006) constitutes a state-of-the-art machine translation service that provides automatic translation among several language pairs including the four Spanish languages plus English, Portuguese and French. See Figure 3. Besides, Opentrad is

[2] http://www.opentrad.com/

Language	Sentence example
English	The entire wealth of the country in its different forms, irrespective of ownership, shall be subordinated to the general interest.
Spanish	Toda la riqueza del país en sus distintas formas y sea cual fuere su titularidad está subordinada al interés general.
Catalan	Tota la riquesa del país en les seves diverses formes, i sigui quina sigui la titularitat, resta subordinada a l'interès general.
Gallego	Toda a riqueza do país nas súas distintas formas e calquera que sexa a súa titularida de está subordinada ó interese xeral.

Table 2. Sentence example from the Spanish Constitution.

designed to be adapted and configured according to user needs, allowing its integration with other systems. Opentrad's design allows for its customization and personalization both from a linguistic point of view, adopting the style book of an organization, and from a technical point of view, allowing its integration into IP networks or a full integration with other systems.



The age of machine translation

Multilingual, open-source and for free

Opentrad is an open-source machine translation system designed initially for all the languages in Spain and for free for everyone.

Currently translation is available between more than twenty language pairs: we can find English or French, as well as Basque and Galician, thus offering a wide range of possibilities.

Besides, Opentrad is designed to be adapted and configured according to user needs, allowing its integration with other systems.

Right now thousands of people use Opentrad every day for the translation of texts, documents or websites. Other users use it to offer their personal or business pages, blogs, etc.. in English, Galician, Spanish, Portuguese, Catalan, Basque, etc.

Customize it

Opentrad's design allows for its customization and personalization both from a linguistic point of view, adopting the style book of an organization, or from a technical point of view, allowing its integration into IP networks or a full integration with other systems.

Eleka Ingeniaritza Linguistikoa and imaxin|software offer different value-added services that allow a total customization and integration of Opentrad, adapting itself to your needs.

¿Quieres Aprender Inglés?
Método Estudio con Tutor Personal
Conversador Nativo y Aula OnLine
www.IdiomasMasterD.com/Ingles

Aprende Inglés y Francés
Aprende idiomas desde casa. Cursos
con Promociones de Lanzamiento!
www.gnanis.es

Solteras fotos anuncios
Solteras con fotos, anuncios videos
Webcam, chat de video. Entra gratis
www.LikeYou.es/Solteras

An interview to Opentrad

15-10-2010

During the SIMO Network 2010 fair that took place in Madrid from the 5th to the 7th of October, Opentrad was interviewed by Mallando na e-ira, a web site that works on the new technologies.
Opentrad is... complete news

Visit Opentrad at the SIMO Network fair

06-10-2010

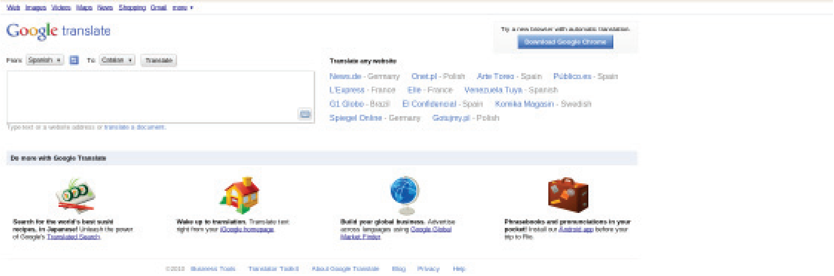
The companies of the Opentrad machine translation consortium, Eleka Ingeniaritza Linguistikoa S.L and imaxin|software, are presenting the new technologic and linguistic developments that offers the... complete news

Fig. 3. Opentrad screenshot

2. A statistical-based system implemented with the Google API translation[3]. See Figure 4. Google's research group has developed its statistical translation system for the language pairs now available on Google Translate. Their system, in brief, feeds the computer with billions of words of text, both monolingual text in the target language, and aligned text consisting of examples of human translations between the languages. Then, they apply statistical learning techniques to build a translation model.

[3] http://code.google.com/apis/ajaxlanguage/

The *detect language* option automatically determines the language of the text the user is translating. The accuracy of the automatic language detection increases with the amount of text entered. Google is constantly working to support more languages and introduce them as soon as the automatic translation meets their standards. In order to develop new systems, they need large amounts of bilingual texts.



Type text or a website address or translate a document.

Search for the world's best sushi recipes, in Japanese! Unleash the power of Google's Translated Search.

Wake up to translation. Translate text right from your iGoogle homepage.

Build your global business. Advertise across languages using Google Global Market Finder.

Phrasebooks and pronunciations in your pocket! Install our Android app before your trip to Rio.

Fig. 4. Google Translate screenshot

The monolingual information retrieval step was implemented by using Solr, which is an XML-based open-source search server based on the Apache-Lucene search library [4]. See Figure 5. Particularly, Solr is the popular, blazing fast open source enterprise search platform from the Apache Lucene project. Its major features include powerful full-text search, hit highlighting, faceted search, dynamic clustering, database integration, and rich document (e.g., Word, PDF) handling. Solr is highly scalable, providing distributed search and index replication, and it powers the search and navigation features of many of the world's largest internet sites.

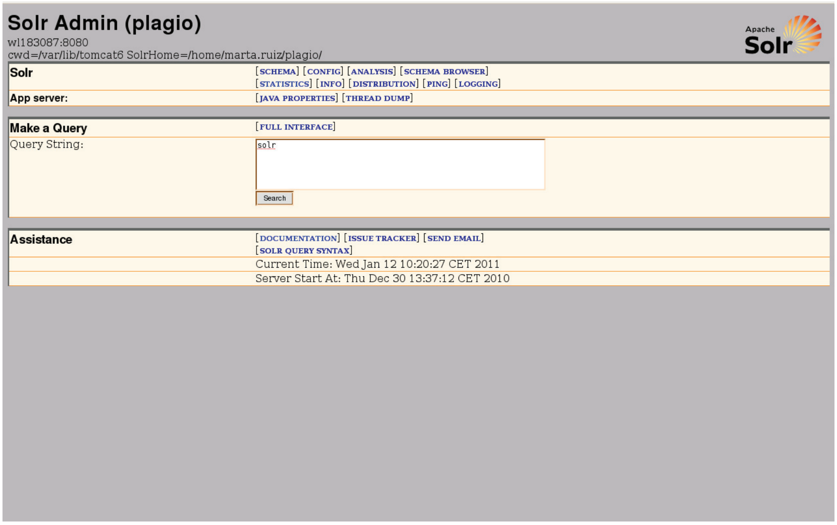
Solr is written in Java and runs as a standalone full-text search server within a servlet container such as Tomcat. Solr uses the Lucene Java search library at its core lecelfor full-text indexing and search, and has REST-like HTTP/XML and JSON APIs that make it easy to use from virtually any programming language. Solr's powerful external configuration allows it to be tailored to almost any type of application without Java coding, and it has an extensive plugin architecture when more advanced customization is required.

Table 3 summarizes the results obtained from the comparative evaluation between the two contrastive systems. We measure the quality of the system in terms of accuracy. We show top-1 and top-5 results. The former reports the percentage of times that the correct result coincides with the top-ranked sentence retrieved by the system and the latter reports the percentage of times that the correct result is within the top-five ranked sentences retrieved by the system.

The query translation system using statistical translation performs slightly better than the rule-based system. It is worth noticing the high quality of cross-language sentence matching using the query translation approach. This high quality is mainly due to the quality of translation.

Figure 6 shows some examples of the system performance.

[4] http://lucene.apache.org/solr/tutorial.html


Apache Solr
[SCHEMA] [CONFIG] [ANALYSIS] [SCHEMA BROWSER]
[STATISTICS] [INFO] [DISTRIBUTION] [PING] [LOGGING]
[JAVA PROPERTIES] [THREAD DUMP]
[FULL INTERFACE]
Search
[DOCUMENTATION] [ISSUE TRACKER] [SEND EMAIL]
[SOLR QUERY SYNTAX]

Solr Admin (plagio)
wl183087:8080
cwd=/var/lib/tomcat6 SolrHome=/home/marta.ruiz/plagio/

Solr

App server:

Make a Query

Query String: solr

Assistance

Current Time: Wed Jan 12 10:20:27 CET 2011

Server Start At: Thu Dec 30 13:37:12 CET 2010

Fig. 5. SOLR screenshot

Source language	System	Target language							
		English		Spanish		Catalan		Gallego	
		top-1	top-5	top-1	top-5	top-1	top-5	top-1	top-5
English	rule-based	100	100	95.0	99.5	92.0	96.0	93.0	96.0
	statistical	100	100	**100**	**100**	**100**	**100**	**97**	**100**
Spanish	rule-based	96.0	99.0	100	100	100	100	**99.0**	**100**
	statistical	**97.5**	**100**	100	100	100	100	96	99
Catalan	rule-based	95.5	99.0	100	100	100	100	93.5	97.0
	statistical	**99**	**99.5**	100	100	100	100	**96**	**99**
Gallego	rule-based	93.5	97.5	**99.5**	**99.5**	83.5	90.5	100	100
	statistical	**97**	**98.5**	97	99	**97.5**	**99**	100	100

Table 3. Comparative results.

8. Correlation between machine translation quality and sentence matching performance

We evaluate the quality of the translation in terms of BLEUPapineni et al. (2001) and PER, see table 4. BLEU stands for Bilingual Evaluation Understudy. It is a quality metric and it is defined in a range between 0 and 1 (or in percentage between 0 and 100), 0 meaning the worst translation (where the translation does not match the reference in any word), and 1 the perfect translation. BLEU computes lexical matching accumulated precision for n-grams up to length four Papineni et al. (2001).

PER stands for Position-Independent Error Rate (PER) and it is computed on a sentence-by-sentence basis. The main difference with WER (Word error rate) is that it does not penalise the wrong order in the translation. WER (McCowan et al., 2004) is a standard speech recognition evaluation metric. A general difficulty of measuring performance lies in the fact that the translated word sequence can have a different length from the reference

Source: Si la moción de censura no fuere aprobada por el Congreso, sus signatarios no podrán presentar otra durante el mismo período de sesiones.
Translation-Google: Si la moció de censura no fos aprovada pel Congrés, els signataris no podran presentar cap més durant el mateix període de sessions.
Retrieval: Si la moció de censura no fos aprovada pel Congrés, els signataris no en podran presentar cap més durant el mateix període de sessions.
Translation-Opentrad: Si la moció de censura no anàs aprovada pel Congrés, els seus signataris no podrán presentar una altra durant el mateix període de sessions.
Retrieval: Si la moció de censura no fos aprovada pel Congrés, els signataris no en podran presentar cap més durant el mateix període de sessions.
Reference: Si la moció de censura no fos aprovada pel Congrés, els signataris no en podran presentar cap més durant el mateix període de sessions.

Source:The Congress may require political responsibility from the Government by adopting a motion of censure by overall majority of its Members.
Translation-Google: O Congreso pode esixir responsabilidade política do Goberno, aprobando unha moción de censura por maioría absoluta dos seus membros.
Retrieval: O Congreso dos Deputados pode esixi-la responsabilidade política do Goberno mediante a adopción por maioría absoluta da moción de censura.
Translation-Opentrad: O Congreso pode requirir responsabilidade política desde o Goberno por adoptar unha moción de censure por maioría total dos seus Membros.
Retrieval: O Congreso dos Deputados pode esixi-la responsabilidade política do Goberno mediante a adopción por maioría absoluta da moción de censura.
Reference: O Congreso dos Deputados pode esixi-la responsabilidade política do Goberno mediante a adopción por maioría absoluta da moción de censura.

Source:O Pleno poderá, con todo, avocar en calquera momento o debate e votación de calquera proxecto ou proposición de lei que xa fora obxecto desta delegación.
Translation-Google: The Chamber may, however, take over at any moment the debate and vote on any project or proposed law that had already been the subject of this delegation.
Retrieval: However, the Plenary sitting may at any time demand that any Government or non-governmental bill that has been so delegated be debated and voted upon by the Plenary itself.
Translation-Opentrad: The Plenary will be able to, however, avocar in any moment the debate and vote of any project or proposición of law that already was object of this delegation.
Retrieval: However, the Plenary sitting may at any time demand that any Government or non-governmental bill that has been so delegated be debated and voted upon by the Plenary itself.
Reference: However, the Plenary sitting may at any time demand that any Government or non-governmental bill that has been so delegated be debated and voted upon by the Plenary itself.

Fig. 6. Examples of the system performance.

word sequence (supposedly the correct one). WER is derived from the Levenshtein distance, working at the word level.

We see that Google translator is better than Opentrad in most translation pairs. It may be possible that Google has part of the Spanish Constitution as training material in its system. However, notice that we did not use directly the Spanish constitution that is available from the website *www.la-moncloa.es*, we had to perform a manual postprocessing to correct some errors in the sentence alignment.

After evaluating the quality of translation we computed correlation coefficients between sentence matching accuracies and translation quality metrics. We found out that some of the

Source language	System	Target language							
		English		Spanish		Catalan		Gallego	
		BLEU	PER	BLEU	PER	BLEU	PER	BLEU	PER
English	rule-based	-	-	20.80	49.14	20.02	51.66	17.49	55.34
	statistical	-	-	44.73	31.38	37.98	36.04	16.75	56.27
Spanish	rule-based	20.92	48.53	-	-	68.76	15.65	**72.57**	**14.56**
	statistical	45.57	31.44	-	-	78.55	11.05	32.90	39.78
Catalan	rule-based	20.95	50.56	70.52	14.89	-	-	**54.81**	**23.81**
	statistical	45.86	30.91	87.59	6.24	-	-	29.16	42.49
Gallego	rule-based	18.67	52.47	**75.85**	**12.60**	**57.71**	**22.31**	-	-
	statistical	30.43	41.52	53.02	26.74	43.53	32.79	-	-

Table 4. Comparative results between translation qualities of used rule-based and statistical systems.

computed correlations were quite high, see table 5. All correlations are significant ($p << 0.05$) except for the cases marked with $*$. There is a slightly high correlation between BLEU and top-1 measure in the statistical case, but it is not maintained in the rule-based case. Research in finding an MT measure which is correlated with CLIR quality or sentence alignment quality was not the objective of this work. However, it may be a nice topic for further research.

	system	top-1	top-5	BLEU
top-1	rule-based	-		
	statistical	-		
top-5	rule-based	95.82	-	
	statistical	76.28	-	
BLEU	rule-based	58.17	39.61*	-
	statistical	74.71	53.53	-
PER	rule-based	-55.24	-36.39*	-99.37
	statistical	-75.03	-50.16	-99.46

Table 5. Correlations coeficients.$*$ marks the non-significant correlations.

9. Conclusions

This chapter presented a cross-language sentence matching application. The proposed approach was a query translation cross-language information retrieval system either using a rule-based or a statistical-based translation system.

We tested the performance of rule-based and statistical systems in a multilingual collection based on the Spanish Constitution.

Results show that the statistical-based system performed slightly better than the rule-based system.

Looking at some examples we saw that the errors in sentence matching were different depending on the kind of translation system we were using, which suggests that a system combination strategy could improve the performance.

We evaluated the translation performance of the rule-based and the statistical-based translation systems. The latter performed better in 12 out of 16 translation pairs.

Finally, we saw that translation quality is correlated with the cross-language sentence matching quality, specially in terms of BLEU and top-1 measures.

10. Acknowledgements

This work has been partially funded by the Spanish Department of Science and Innovation through the *Juan de la Cierva* fellowship program.
The authors also want to thank Barcelona Media Innovation Center for its support and permission to publish this research.

11. References

Banchs, R. E. & Costa-jussà, M. (2009). Extracción crosslingue de documentos usando mapas semánticos no-lineales, *SEPLN* 43: 169–176.

Bangalore, S. & Riccardi, G. (2000). Finite-state models for lexical reordering in spoken language translation, *Proc. of the 6th Int. Conf. on Spoken Language Processing, ICSLP'02*, Vol. 4, Beijing, pp. 422–425.

Berger, A. L., Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., Gillett, J. R., Lafferty, J. D., Mercer, R. L., Printz, H. & Ureš, L. (1994). The candide system for machine translation, *HLT '94: Proceedings of the workshop on Human Language Technology*, pp. 157–162.

Braune, F. & Fraser, A. (2010). Improved sentence alignment for symmetrical and asymmetrical parallel corpora, *Coling*, pp. 81–89.

Casacuberta, F. (2001). Finite-state transducers for speech-input translation, *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU*, Trento, pp. 375–380.

Chen, J. & Bao, Y. (2009). Cross-language search: The case of google language tools, *First Monday* 14(3-2).

Chen, S. F. & Goodman, J. T. (1998). An empirical study of smoothing techniques for language modeling, *Technical report*, Harvard University.

Codina, J., Pianta, E., Vrochidis, S. & Papadoupoulos, S. (2008). Integration of semantic, metadata and image search engines with a text search engine for patent retrieval, *Proceedings of ESWC 2008*, Tenerife, Spain.

Costa-jussà, M. & Fonollosa, J. (2009). An ngram-based reordering model, *Computer Speech and Language* 23(3): 362–375.

de Gispert, A., Gupta, D., Popovic, M., Lambert, P., MariÃśo, J., Federico, M., Ney, H. & Banchs, R. (2006). Improving statistical word alignments with morpho-syntactic transformations, *Proc. of 5th Int. Conf. on Natural Language Processing (FinTAL'06)* pp. 368–379.

de Gispert, A. & Mariño, J. (2002). Using x-grams for speech-to-speech translation, *Proc. of the 7th Int. Conf. on Spoken Language Processing, ICSLP'02*, Denver, pp. 1885–1888.

Gale, W. A. & Church, K. W. (1993). A program for aligning sentences in bilingual corpora, *Computational Linguistics* 1(19): 75–102.

González, A. O., Boleda, G., Melero, M. & Badia, T. (2005). Traducción automática estadística basada en n-gramas, *Procesamiento del Lenguaje Natural, SELPN* 35: 69–76.

Kishida, K. (2005). Technical issues of cross-language information retrieval: a review, *Cross-Language Information Retrieval* 41(3): 433–455.

Langlais, P. & Patry, A. (2007). Translating unknown words by analogical learning, *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 877–886.

Mariño, J., Banchs, R., Crego, J., de Gispert, A., Lambert, P., Fonollosa, J. & Costa-jussà, M. (2006). N-gram based machine translation, *Computational Linguistics* 32(4): 527–549.

McCowan, I., Moore, D., Dines, J., Gatica-Perez, D., Flynn, M., Wellner, P. & Bourlard, H. (2004). On the use of information retrieval measures for speech recognition evaluation, *IDIAP-RR 73*, IDIAP, Martigny, Switzerland.

Moore, R. C. (2002). Fast and accurate sentence alignment of bilingual corpora, *AMTA*, pp. 135–144.

Nießen, S. & Ney, H. (2004). Statistical machine translation with scarce resources using morpho-syntactic information, *Computational Linguistics* 30(2): 181–204.

Och, F. (2003). Minimum error rate training in statistical machine translation, *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics*, Sapporo, pp. 160–167.

Papineni, K., Roukos, S., Ward, T. & Zhu, W.-J. (2001). Bleu: a method for automatic evaluation of machine translation, IBM Research Report, RC22176.

Potthast, M., Stein, B., Eiselt, A., BarrÃşn, A. & Rosso, P. (2009). Overview of the 1st international competition on plagiarism detection, *Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse*.

Ramírez-Sánchez, G., Sánchez-Martínez, F., Ortiz-Rojas, S., Pérez-Ortiz, J. A. & Forcada, M. L. (2006). Opentrad apertium open-source machine translation system: an opportunity for business and research, *Proceeding of Translating and the Computer 28 Conference*.

Senrich, R. & Volk, M. (2010). Mt-based sentence alginment for ocr-generated parallel texts, *AMTA*, Colorado.

Soerfel, D. (2002). Thesauri and ontologies for digital libraries, *Proceedings of the Joint Conference on Digital Libraries*.

Tillmann, C. & Ney, H. (2003). Word reordering and a dynamic programming beam search algorithm for statistical machine translation, *Computational Linguistics* 29(1): 97–133.

Utiyama, M. & Tanimura, M. (2007). Automatic construction technology for parallel corpora, *Journal of the National Institute of Information and Communications Technology* 54(3): 25–31.

Varga, D., Németh, L., Halácsy, P., Kornai, A., Trón, V. & Nagy, V. (2005). Parallel corpora for medium density languages, *RANLP*, pp. 590–596.

Vidal, E. (1997). Finite-state speech-to-speech translation, *Proc. Int. Conf. on Acoustics Speech and Signal Processing*, Munich, pp. 111–114.

Zhang, Y., Zens, R. & Ney, H. (2007). Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation, *Proc. of the Human Language Technology Conf. (HLT-NAACL'06):Proc. of the Workshop on Syntax and Structure in Statistical Translation (SSST)*, Rochester, pp. 1–8.

3

The BBN TransTalk Speech-to-Speech Translation System

David Stallard et al.,*
Raytheon BBN Technologies,
USA

1. Introduction

Portable translation devices which enable people who speak different languages to communicate with each another in voice are likely to have a far-reaching impact in both the civilian and military worlds. While long a staple of science fiction, a la the "Star Trek universal translator", devices that actually translate between languages have not been fully realized. In the last decade, however, under the sponsorship of the Babylon and TRANSTAC (Translation for Tactical use) programs of the US Defense Advance Research Program Agency (DARPA), several research sites, including BBN (Stallard et al., 2007), CMU (Waibel et al., 2003), IBM (Gao et al., 2006), SRI (Akbacak et al., 2009) and USC (Belvin et al. 2005), have made significant progress in developing a real two-way speech-to-speech (S2S) translation systems. These systems are not “universal translators” in the science-fiction sense, in that must be configured for the language and conversational domain of interest, rather than spontaneously understanding them. However, the technology is language-independent, and under the auspices of the TRANSTAC program, systems have been configured for several different foreign languages of interest to the US Government, including Iraqi Arabic, Malay, Farsi, Dari, and Pashto. Though the technology is also domain-independent, most of these systems support conversations in the so-called "force protection" military domain, which is broadly construed to include not only conversations relevant to checkpoints, searches, and other military operations, but also rapport building, civil affairs, and basic medical conversations.

In this article, we describe BBN’s S2S system, TransTalk, which runs not only on laptops and ultra-mobile PCs, but also on mobile Android Smartphones, running locally on the device itself and not a server. In common with other TRANTSTAC systems, TransTalk’s technology is language-independent, and has been configured to translate between English and numerous other languages, including Iraqi Arabic, Dari, Pashto, Farsi, and Malay. TransTalk also has a uniquely simple user interface which does not require the user to view a screen. TransTalk integrates automatic speech recognition (ASR), machine translation (MT), and text-to-speech (TTS) synthesis engines for converting speech in one language into a different language. In particular, TransTalk uses the BBN Byblos ASR engine for converting speech to text. BBN Byblos is a multi-pass, speaker-independent large

*Rohit Prasad, Prem Natarajan, Fred Choi, Shirin Saleem, Ralf Meermeier, Kriste Krstovski, Shankar Ananthakrishnan and Jacob Devlin vocabulary speech recognizer which uses n-gram language models instead of a finite state grammar. For machine translation, TransTalk primarily uses a BBN-developed Statistical Machine Translation (SMT) component. For text-to-speech synthesis, we use engines developed by TTS research sites under the DARPA TRANSTAC program, which includes CMU and Cepstral. TransTalk has consistently been a top performer in independent applications conducted by the US government.

Of particular importance to the recent progress in S2S technology has been the adoption of Statistical Machine Translation (SMT). SMT uses a statistical, corpus-driven approach, rather than hand-coded translation rules; and is driven by automated rather than manual performance evaluation. There are two advantages conferred by SMT. First, the statistical paradigm generally provides better performance than approaches based on hand-written rules, as these rules are often brittle and conflict with one another. Second, the automated nature of the process allows much more rapid development and testing of new approaches to improve performance. The resulting labor savings has greatly accelerated the progress of both machine translation and S2S as a whole. In this way, SMT may be seen as following in the footsteps of ASR, which also underwent dramatic improvement following the adoption of statistical paradigms and automated evaluation.

S2S systems are configured for particular language pairs by training ASR and translation models using speech and language data (recordings, transcriptions, and translations) in the relevant languages. For optimal performance, the data collected should match the intended domains of conversation of the system. That is, if the intended domain is force protection, data should be collected for that domain. While data outside that domain can be helpful for general modeling of the given language, it often lacks the key concepts and constructions that are important in the specific application domain. In practice therefore, the domain-relevant data is frequently collected in simulated translingual dialogs between role-playing individuals. Because of the finite resources available for such data collection, and because many of the languages of interest are low-resource themselves, S2S technology must frequently cope with sparse data, which poses challenges for both ASR and MT.

In this paper, we describe the individual components of our system, including its ASR, MT, TTS, dialog manager, and user interface components, and their integration into a free-form two-way S2S translation system. We present novel algorithms for overcoming specific challenges posed by colloquial and low resource nature of the languages of interest. Another important issue in speech-to-speech translation is using some form of confirmation strategy for minimizing errors in transferring a concept from one direction to other. Such errors can easily cause the dialog to drift or stall. Here, we present multiple confirmation techniques for a user to get feedback from the system so as to detect errors in the concepts being conveyed by the system. In addition, we describe a novel methodology for assessing the usefulness of these user confirmation strategies.

The remainder of the paper is organized as follows. Section 2 gives a brief overview of the our TransTalk system. Section 3 discusses our user-centered approach to system design and interaction. Section 4 presents work we have done on our ASR component, with particular emphasis on improving performance for colloquial dialects of low-resource languages. Section 5 discusses the machine translation component of our system, and continues the emphasis on low-resource languages. Section 6 presents live evaluation results for our system, and Section 7 concludes.

2. System overview

A block diagram of the BBN TransTalk system is shown in Figure 1. The BBN TransTalk system uses BBN's Byblos speech recognizer (Nguyen and Schwartz, 1997), BBN's SMT engine, and third-party text-to-speech synthesizer(s). Various input modalities are supported, including both handheld and headset microphones. The primary physical interface is the "BBN SuperMic", a handheld unit developed by BBN, which encompasses a directional microphone, speakers, and two push-and-hold "listen" buttons, one for receiving the English speech, and the other for receiving the foreign speech. Figure 2 shows the BBN TransTalk system running on multiple platforms: (a) Ultra-Mobile PC (UMPC) with BBN SuperMic, and (b) Android smartphone.

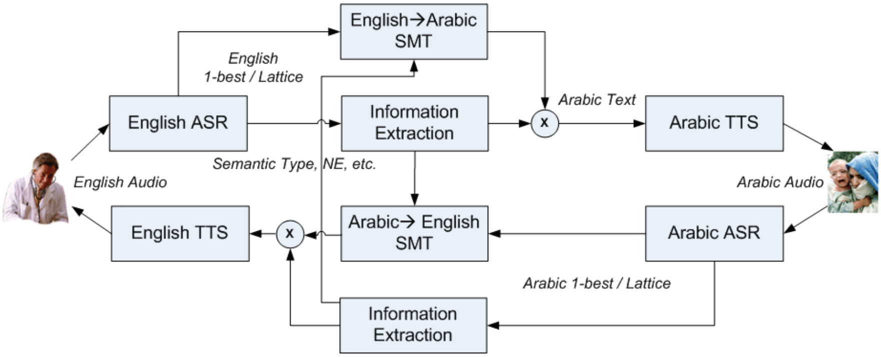


Fig. 1. Block Diagram for BBN TransTalk 2-Way S2S Translator.

English speech received through the physical interface device is sent to the English speech recognizer, which outputs a sequence of words in the recognizer's vocabulary. This English text is then sent to both the SMT component and to a separate information extraction component. The information extraction component performs the following functions:

1. English speech received through the physical interface device is sent to the English speech recognizer, which outputs a sequence of words in the recognizer's vocabulary. This English text is then sent to both the SMT component and to a separate information extraction component. The information extraction component performs the following functions:
2. Canonicalization: matches the utterance to one of a set of utterances for which it has stored translations (Stallard et al., 2007). If none is found, the output of the SMT component is used instead. Arabic speech corresponding to the translation is then played back. This may be a pre-recorded wave file, or more generally, the result of text-to-speech synthesis.
3. Question Detection: determines whether the recognized utterance is a question or a statement. This module also classifies the question as one of the pre-defined classes.}
4. Named Entity Detection: detects whether the spoken response has named entities such as person names, place names, geo-political organizations, etc. (Prasad et al., 2008; Bikel et al., 1999).

A composite foreign language translation ("Arabic") in Figure 1 is produced by the SMT and the information extraction component. This translation is then either played as a pre-recorded wave file, or more generally, by the text-to-speech synthesis.

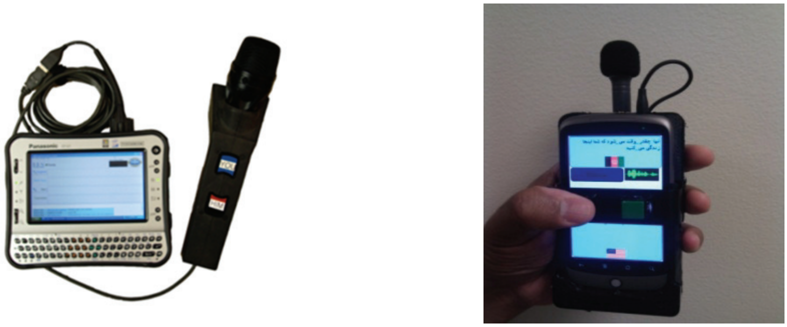


Fig. 2. BBN TransTalk Two-way S2S Translator on UMPC and Smartphone platform.

The foreign language speaker's reply ("Arabic" in Figure 1) is sent to the foreign language recognizer, which outputs text in the foreign language. The foreign language text is translated into English text with a process similar to the one in English-to-Foreign direction. The translated text is then sent to a second speech synthesizer, which speaks it out for the English speaker to hear, and/or displaying it on a screen.

3. User-centered interaction

3.1 Overall design

A key aspect of our TransTalk system is its user-centered interface. Our design of the interface was guided by a number of desiderata. Obviously, the interface had to be simple, easy to use, and efficient. But it also had to work without a screen display, so that it could be used by soldiers in “eyes-free” operation. We also wanted it to be easy for the user to detect when the system had made an error in translation, and to correct the error. Finally, we wanted the user to be able to abort system output and barge in at any time to speak again, no matter what the system was doing at the time. The user interface design we developed provides an elegant joint solution to all of these goals.

In physical terms, the system’s user interface is indeed simple. It consists of just two push-and-hold buttons, one labeled “YOU”, the other “HIM”. The YOU button commands the system to begin listening for English speech, and performing ASR on it as soon as speech is heard. The HIM button, similarly, causes the system to begin listening for speech in the foreign language and performing ASR on it. When the button is released, the system stops listening, finishes up ASR, and then passes the ASR output to MT for translation into the opposite language, and then to TTS for speaking out to the other party.

The requirement that the ELS be able to detect translation errors has impact on the interface’s behavior. It is simply a fact that despite ongoing improvements in the underlying technologies, for the foreseeable future, S2S systems will make errors in translation. If undetected, translation errors can lead to mutual incomprehension and a complete breakdown of the dialog. To cope with this problem, our system presents the ELS with a “confirmation” utterance that tells him what the system thought he said, so that he can determine whether the system made an error. In the system’s usual mode of operation, this confirmation utterance is simply a read-back of the English ASR result. (We discuss alternatives such as “back-translation” in a later section).

Now, if a display screen were available, this confirmation utterance would simply be displayed on the screen and the ELS could quickly scan it to determine whether he had been understood correctly. However, the TRANSTAC program requires (and most military users prefer) that the system be usable without a display. The only way the confirmation utterance can be delivered is through voice. One possibility would be for the system to explicitly ask the user via TTS "Did you say 'Show me your id'?". However, this "explicit confirmation" would slow down the dialog, and the repeated confirmation interactions would likely be irritating to the user. Our system instead uses "implicit confirmation", in which it speaks out the confirmation utterance for the ELS to hear. If the ELS decides this is correct, he takes no action, and the system's processing continues as normally, generating the translation and playing it out for the foreign language speaker (FLS) to hear. If the ELS instead decides that the confirmation is incorrect, he simply presses the "YOU" button again. This aborts any ASR, MT, or TTS activities the system may be performing, and in particular halts voice output in either English or the foreign language. The system then begins listening to the ELS, who may speak again, either repeating his utterance more clearly, or rephrasing it, as he chooses.

Because it aborts all ongoing system activities, the "YOU" button effectively doubles as an "abort" button. If the ELS wants to abort the system's current activities, but does not want to speak again right away, he can simply press the YOU button and then quickly release it again, without speaking. The system recognizes this very short listening interval as being an abort, rather than a speech event, and ignores the empty result that ASR returns. In this way, we avoid the need for a dedicated third "abort" button, thereby retaining our maximally simple two-button interface.

The above-described abort and barge-in functionality of the YOU button illustrates another key design goal of our system, which might be stated as "The user controls the system; the system does not control the user". That is, the system does not constrain the ELS user's actions, but rather allows him to interrupt it at any time, and speak again without having to wait for the system to be "ready". He can simply assume that the system is always ready.

Such a capability is not straightforward to achieve, however. In the synchronous pipeline of ASR, MT, and TTS, the various components can be in different states when the abort/barge-in occurs. Example states include processing the last input, returning results for the last input, aborting in response to the button push, and resetting internal data structures to prepare for the next input. A component that is in any of these states will not be ready to begin immediately processing new input. The easiest way to cope with this might be to require all system components to return to their ready state before allowing new input from the user (perhaps using a beep as a ready signal). However, the time that it would take all the system's components to return to their ready state is variable, and depends among other things upon which component(s) were interrupted, and which state they were in when interrupted. To force the ELS to wait until all components are in their ready state before he speaks again, even by 10's of milliseconds, is to invite user frustration and user error. In particular, it would be very difficult to prevent the user from speaking too early, thus resulting in truncated utterances being sent to the ASR, with the consequent loss of speech recognition and translation accuracy.

Our approach avoids these problems. Instead of attempting to configure the user's behavior to cope with the situation, we configure the system's internal behavior. That is, if a component is not yet ready for new input, the system buffers the input until the component returns to the ready state and can begin processing it. Examples include user speech (for ASR), English ASR output (for MT), and foreign-text MT output (for TTS).The system begins to buffer user speech, in particular, as soon as the button is pressed. Any slight latency that may results from this internal wait will be manifested only as a slight delay in the system's final output, which will probably not be noticeable by the user, rather than a delay enforced on the user's input, which would certainly be noticeable by him. The overall theme of our interface is that the system retains its internal complexity inside itself, where it belongs, rather than imposing it upon the user, who has more important things to do.

3.2 Improving the efficiency of voice confirmation

An obvious efficiency issue presented by voice confirmation is the additional time it costs the interaction. Confirming the ELS's utterance means that each English utterance is spoken twice, first by the ELS, and then again by the system. To alleviate this, we can substantially reduce the effective time that confirmation costs by performing the E2F MT concurrently with the confirmation TTS. Since the E2F MT would have to be performed anyway whether or not confirmation was done, doing it at the same time as confirmation, rather than waiting until confirmation is done, saves time. Effectively, we are reducing the time that confirmation TTS costs the dialog (avg. 3.0 seconds), by subtracting the time that the MT takes to run (avg.1.2 seconds), yielding a relative reduction of 40%. Figure 3 illustrates this time savings, with the top panel representing confirmation and MT running in series, compared with confirmation and MT running in parallel.

Moreover, if the ELS and FLS are listening on separate channels, as is the case in the two-phone configuration, we can obtain even greater time savings by also playing the Iraqi translation TTS itself at the same time as the English confirmation TTS. In this mode, the system begins playing the foreign TTS as soon as the E2F MT produces the foreign-language utterance for it. Given that E2F is faster than confirmation, it will generally be the case that the system will then be speaking English and foreign-language utterances simultaneously. However, because each party has his own phone, neither hears the TTS output intended for the other. Because the system is speaking in two different languages in parallel, we term this technique "parallel confirmation".

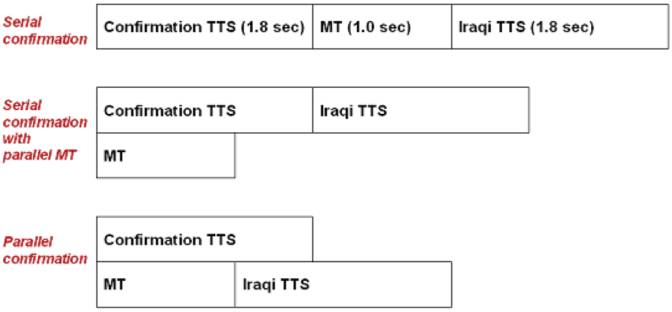


Fig. 3. Different Confirmation Modes.

Figure 3 illustrates the benefits of this technique by graphically comparing Serial Confirmation, Serial Confirmation with Parallel MT, and Parallel Confirmation. Note that Parallel Confirmation effectively reduces the time-cost of confirmation to zero, since in this configuration the confirmation is performed entirely in parallel with activities that would need to be performed anyway, and whose combined duration exceeds that of the confirmation.

Note also an additional important benefit of parallel confirmation; namely, that it enables improvements in the MT's speed to increase the system's overall translation speed, specifically, by reducing the lag before foreign TTS starts. By contrast, in Serial Confirmation with Parallel MT, no further throughput increase is possible, once the time that the MT takes to translate the ELS's English utterance is less than the time that the English confirmation TTS takes to speak that utterance.

We have used the parallel confirmation effectively in a two-phone configuration, where each speaker has his own handset. The two phones communicate via Bluetooth and send text back and forth, allowing confirmation TTS and translation TTS to be generated in parallel on the respective handset.

3.3 Generating the confirmation utterance

As stated previously, our primary method of generating a confirmation utterance is to simply read back the ASR output. The advantage of this approach is that it allows the user to catch ASR errors quickly, which is important since ASR errors on concept words guarantee a wrong translation. The disadvantage, of course, is that correctness of the ASR output says nothing about whether the MT output itself was correct. In particular, it is perfectly possible for the ASR output to be error-free, yet for the MT result for that output to be completely wrong.

An alternative strategy that addresses this issue is "back-translation", in which the output of ASR + MT into the target language is translated back into the source language, and the result played back to the user for his approval. A back-translation that is close to the original utterance in meaning can have the important psychological advantage of making the user feel more secure that he was translated correctly. In fact, in informal interviews, users of our S2S system who have use system with the back-translation alternative to cofirmation express a strong approval of it.

The back-translation approach can be objected to, however, on two grounds.The first objection is that the output of back-translation, having been passed through three successive noisy channels (ASR, forward MT, and backward MT), will likely be hopelessly garbled, causing most forward translations to appear wrong. The second objection is that even when not garbled, back-translation may yet be misleading, since the same incorrect phrase pair rule used in the forward direction may also be selected (in reversed form) in the backward direction, leading back to the original source phrase again and leaving the error undetected.

The most straightforward way to evaluate the efficacy of back-translation and other confirmation approaches would be to run two complete sets of live evaluations, one with the approach in question and one without, and compare the results on measures such as concept transfer, rate of concept transfer, user satisfaction, and the like. Unfortunately, given the great expense of carrying out realistic evaluations, which involve assembling personnel from multiple locations in the US, this is infeasible. We therefore must look for some offline method of evaluating back-translation.

The implicit hope of back-translation, and indeed of any other confirmation utterance-based strategy, is that the ELS could develop a mental model, based on his experience in using the system, of the minimal level of back-translation quality that would predict that the forward translation is correct. We do not concern ourselves here with how the ELS would develop such a mental model, but rather seek to determine whether it is even possible to develop such a model at all – that, to determine whether the data is sufficiently consistent that is possible to infer useful prediction rules from it.

As a subjective measure of translation quality in either direction, we use the familiar 1-5 Likert scale to rank both forward translations (i.e. English-to-foreign), and back-translations. We do not assume that users will actually assign Likert scores while using the system; but instead view the score as a numerical proxy for the user's reaction. We assign the following interpretations to the different elements of the Likert scale.

5: Essentially a perfect translation.
4: An adequate if slightly disfluent translation which conveys the utterance's meaning
3: A partial translation which is missing one or more concepts, or is severely disfluent.
2: A translation which is missing most of the concepts.
1: A translation with no apparent relation to the input.

If back-translation were a perfectly effective diagnostic, the Likert rating of the back-translation and the Likert rating of the forward translation would have the same value. Obviously, this will seldom be the case, since both are noisy processes, with one of them operating on the output of the other. One might then fall back to a weaker requirement, for example only requiring that the back-translation and forward-translation quality be well-correlated in a linear relationship.

Our approach to this problem is quite different. Note that we are not interested in predicting the actual value of the Likert rating for the forward translation, but rather in simply predicting whether or not the forward translation's Likert rating is above a certain threshold of acceptability. Therefore, we seek to use the back-translation for binary classification, rather than regression.In particular, we choose a specific minimum acceptable Likert score *F* for the forward translation - say, a score of 4. We then test various minimum thresholds *B* for the back-translation Likert score. In particular, for utterances whose back-translation score is at or above the threshold *B*, we test the prediction that the utterance's forward translation Likert score will be at or above the threshold *F*, and thus acceptable. Below *B*, we predict that the forward translation Likert will be below *F*, and therefore unacceptable. We compute precision, recall, and F-measure for each such threshold. Different costs for the different error types, e.g. a higher penalty for false acceptance than for false rejection, can be straightforwardly taken into account by using a weighted harmonic mean.

To test the evaluation methodology outlined above, we used a set of 779 English utterances that were spoken to our system by ELS users during the TransTac live evaluation in June 2008, conducted by the US government's National Institute of Standards and Technology (NIST). In this evaluation, active-duty military personnel played the part of the ELS, while native speakers of Iraqi Arabic were recruited to play the part of the FLS. The utterances of both parties, and the system's ASR and MT outputs for these, were recorded for later analysis. The Iraqi translation output produced by the system for the ELS's utterances was Likert-scored by a native Arabic speaker experienced in the application domain. To produce the back-translations, we ran (offline) our Arabic-to-English MT on the system's Iraqi translation outputs. These back-translations were then Likert-scored by a native English speaker knowledgeable in the application domain. For comparison, the same English ranker also Likert-scored the output of our system's English ASR for these same 779 utterances (i.e. the ASR read-back strategy).

The resulting scores are shown in Table 1. As can be expected, the highest mean Likert scores were produced on ASR output, which tends to overestimate the true (forward) Likert score, while the lowest were associated with back-translation output, which tends to underestimate it. Both were approximately equally well-correlated with forward Likert score, however, with a correlation coefficient of approximately 0.60. The English ASR WER obtained on this corpus was 6.2%, while the English-to-Arabic BLEU score on this ASR output was 56.7%.

Some examples of back-translations and their Likert rankings are: "Turn off your vehicle" (for "Turn your vehicle off"), ranked 5; "Construction prior experience do you have" (for "Do you have prior construction experience"), ranked 4; and "How many subcontracting work" (for "How many subcontractors work for you") ranked 3. Table 1 shows the mean Likert scores for each of the conditions, namely, forward translation, back-translation, and ASR output of Likert scores for the back-translation.

To obtain results on back-translation efficacy, we set the forward translation Likert score threshold F to be 4.0. This may be considered a good minimum acceptable score for our purposes, as scores below 4.0 are by definition associated with "semantic damage" to the translation. Table 2 gives acceptance rate, false rejection rate, false acceptance, F-measure, and precision-weighted F-measure for different back-translation Likert score cutoffs *B*. Each row of this table can be interpreted as a prediction rule, which predicts that an utterance whose back-translation Likert score is at or above the cutoff will have a forward translation whose Likert score will be 4.0 or higher.

Forward-trans.	Back-trans.	ASR
4.42	3.99	4.64

Table 1. Mean Likert Scores.

For many S2S applications, a false acceptance can be regarded as worse than a false rejection, because of the possibility of confusing the respondent, etc. For example, one might decide that a false acceptance is twice as bad as a false rejection. The rightmost column of Table 2 gives F-measure computed with these weights (0.67 vs. 0.33).

The results in Tables 1 and 2 seem to show that the worst fears regarding back-translation are not realized. By no means does back-translation yield incomprehensible utterances for most cases, nor is it a false and over-optimistic guide. Indeed, for cutoffs of 4.0 or higher, its false acceptance rate is actually quite low. This precision does come at the expense of recall, however, and in particular at a cutoff of 4.0 fully 39% of ELS utterances would be rejected and have to be retried. A better strategy might be a slightly less strict cutoff of 3.5, which yields a low false acceptance rate of 8%, while falsely rejecting only 14%. This rule corresponds to a back-translation which subjectively seems rather poor, but which is not completely deficient.

Cutoff	Acpt	FlsRej	FlsAcc	FMsr	WFMsr
5.0	0.27	0.68	0.02	0.48	0.58
4.5	0.35	0.59	0.03	0.57	0.67
4.0	0.61	0.29	0.04	0.81	0.86
3.5	0.78	0.14	0.08	0.89	0.90
3.0	0.94	0.02	0.14	0.92	0.90
2.5	0.96	0.02	0.15	0.91	0.89
2.0	0.99	0.00	0.16	0.91	0.88
1.0	1.00	0.00	0.17	0.91	0.88

Table 2. Precision and Recall for Back-translation.

The key question to be addressed, however, is whether back-translation is better than our default confirmation strategy of simply reading back the system's English ASR output. To address this question, Table 3 repeats the above experiment, using Likert rankings on the system's English ASR output. Note the false acceptance rate is higher than for back-translation, but the false rejection rate is much lower, yielding good F-measure scores at all values of the cutoff. For most cutoff values, the ASR read-back strategy even slightly out-performs back-translation on weighted F-measure.

Cutoff	Acpt	FlsRej	FlsAcc	FMsr	WFMsr
5.0	0.72	0.19	0.07	0.86	0.88
4.5	0.75	0.17	0.08	0.88	0.89
4.0	0.86	0.07	0.10	0.92	0.91
3.5	0.94	0.02	0.13	0.92	0.90
3.0	0.99	0.00	0.16	0.91	0.89
2.5	0.99	0.00	0.16	0.91	0.89
1.0	1.00	0.00	0.17	0.91	0.89

Table 3. Precision and Recall for ASR Readback.

It might seem from this analysis that ASR read-back is a superior strategy to back-translation. It should be noted, however, that ASR read-back on this dataset has a floor of 7% false acceptance, below which it cannot possibly go. The back-translation strategy, by contrast, can go as low as 2% false acceptance, albeit at the price of a very high false rejection rate. If the goal were to fix a certain maximum allowable rate of false acceptance rate – say 8% - rather than maximizing F-measure, the back-translation strategy could be seen as slightly superior, resulting in a 14% false rejection rate as opposed to ASR read-back's 17%.

4. Automated Speech Recognition (ASR)

4.1 Overview

The ASR component of our S2S system is the BBN Byblos speech recognizer (Nguyen and Schwartz, 1997). Byblos models speech as the output of context-dependent phonetic Hidden Markov Models (HMMs). The outputs of the HMM states are mixtures of multi-dimensional diagonal Gaussians. Different forms of parameter tying are used in Byblos, including State Tied Mixture (STM) triphone and State Clustered Tied Mixture (SCTM) quinphone models. The mixture weights in both these cases are shared based on decision tree clustering using linguistic rules. Decoding is performed using our patented two pass search strategy (Nguyen and Schwartz, 1997). The forward pass is a fast-match beam search using an STM acoustic model and an approximate bigram language model. The output of the forward pass consists of the most likely word-ends per frame along with their partial forward likelihood scores. The backward pass operates on the set of choices from the forward pass to restrict the search space, and uses the more detailed SCTM quinphone model and a trigram language model to produce the best hypothesis.

Development of ASR capability for our S2S system posed special challenges, as many of the languages of interest, including Iraqi Arabic, Pashto, and Dari ,are not only low-resource, but also of colloquial dialect. Such languages are challenging for ASR development for two reasons. First, in many cases there is no standard written form for the colloquial dialects of a language, leading to a lack of consistency in transcriptions of audio in that language. Second, creating pronunciation lexicons for words in these dialects is challenging. Most ASR engines use phones as units for acoustic modeling, and each word in the recognition lexicon is manually spelled using these phones. Given that skilled acoustic-phoneticians for low-resource languages are few, the manual creation of phonetic spellings for large vocabulary ASR in low-resource languages is generally impractical. Moreover, creating a phonetic dictionary is even more difficult for languages that use the Arabic script for their writing system. This is because in most of the colloquial dialects of such languages (e.g. Iraqi Arabic, Farsi, Dari, and Pashto), short vowels do not correspond to characters of their own, but instead appear as diacritic marks on other characters, and furthermore, are usually omitted. This results in additional pronunciation ambiguity and language-model confusability for vowel-less word forms which may correspond to several different actual words. A a classic example is the Arabic root form having the meaning of writing or inscribing, "k-t-b", which can appear with many different vowel forms, some of which correspond to the "book", "writer", "he wrote", "bookdealer", etc. Nevertheless, all these forms are typically written simply as "ktb".

Given these challenges, most state-of-the-art ASR systems resort to the "grapheme-as-phoneme" approach for lexicon creation (Billa et al., 2002).In this approach, the pronunciation for a word is derived directly from the orthography by treating the constituent character/grapheme as phones. The grapheme approach has several advantages including: (1) it automates the dictionary creation process, thereby simplifying the ASR training, (2) it does not suffer from inter-annotator differences in manual pronunciation creation for words, and (3) it allows the automated addition of new vocabulary at runtime.

While the grapheme-as-phoneme approach has emerged as a promising approach for mitigating the impact of inherent ambiguity introduced by absence of short vowels, researchers have also explored automatic diacritization based on morphological analysis (Xiang et al., 2006). However, such automatic diacritization methods have several shortcomings, - most of which are due to the creation of large number of vowelization variants, of which very few are actually useful. The increased number of pronunciation variants for a given word has several undesirable effects. First, it typically increases the word's confusability with other words, because the difference in pronunciation between words usually becomes smaller. Second, it increases the search space during decoding, because the decoder has to consider a larger number of pronunciations for each word. Finally, most of the rules used by morphological analyzers for a given language were developed for the language's formal form, and tend to break down when applied to colloquial dialects. Therefore, the grapheme approach is usually still better than using automatic diacritization for colloquial dialects. Nevertheless, the recognition performance with grapheme-as-phonemes is significantly worse than with a high-quality, manually created phonetic dictionary.

In this section, we present techniques that reduce the difference between grapheme and full phonetic systems by using manual pronunciations for only a small fraction of words (Prasad et al, 2010). Specifically, we investigate two different techniques for developing a recognizer for colloquial Pashto. The first technique uses a modified version of the text-to-phoneme (T2P) tool (Black et al., 1998). T2P is a decision tree approach that learns letter-to-sound rules from a small set of manual pronunciations. The standard version of T2P has serious limitations for languages for which the number of letters/graphemes is significantly different from the number of phones. Here, we describe a novel approach for extending T2P to deal with such languages. The second technique uses a hybrid phoneme/grapheme recognition approach, similar to the one described in (Magimai-Doss et al., 2004).

4.2 Automated lexicon creation

Grapheme-as-Phoneme: We developed a grapheme-as-phoneme (Billa et al., 2002) mapping based on the orthography of the words in the Pashto data in the TRANSTAC corpus. The Pashto corpus spans a wide range of scenarios, including checkpoint patrols, civil affairs, medical interviews, facility inspections, etc. The audio in the corpus was segmented and transcribed by Appen, Pty, Ltd. We first pre-processed Appen's audio data and transcriptions in order to eliminate segments with transcriptions that are not suitable for either acoustic or language model training, e.g. unintelligible speech, long pauses, overlapping, or foreign speech. Next, we divided the speakers and data into two sets: A 34-hour (400K total, 10K unique words) training set and a 2-hour, 26K total word, test set.

We used a modified Buckwalter transliteration system to create Romanized forms of Pashto letters. A total of 34 phones were derived from the graphemes after Romanization. Because several letters map to the same sounds, the total number of graphemes is less than the total number of letters in Pashto alphabet. In Table 4, we compare the phone set used for the phonetic and grapheme representations.

Representation	Pashto Sounds	Non-speech	Total Phonemes
Phonetic	42	3	45
Grapheme	34	3	37

Table 4. Pashto phoneme and grapheme representation

Learning Text-to-Phoneme Mappings: Our approach for text-to-phoneme conversion is based on the set of public-domain tools from CMU (Black et al, 1998). The training of T2P models with the CMU tools is performed in three steps:

1. Align letters to phonemes in the training dictionary
2. Extract contextual features from the alignments
3. Train a decision tree using the contextual features.

We found serious limitations in the alignment step of the standard T2P tool. Specifically, the standard alignment process can only handle word and pronunciation pairs where the number of letters is greater or equal to the number of phones, allowing no more than one phone to be aligned to a given letter. While this may be acceptable for most of English words, it does not work for many other languages including Pashto.

Therefore, we implemented a new alignment algorithm that overcomes the limitations of the standard T2P tool. The algorithm uses iterative expectation maximization (EM) style optimization to find alignments that best describe the training dictionary. Our updated alignment algorithm has the following key steps:

1. Initialization
   a. Set P(phone = p | letter = c) = $\frac{num\ dictionary\ pairs\ with\ both\ c\ and\ p}{num\ words\ with\ c}$
   b. Set P(deletion | c) = 0.1
2. Iterate until convergence
   a. Find best path (according to the current model) through [letters phones] grid using dynamic programming and allowing any number of phones per letter
   b. Update P(p | c) = $\frac{number\ of\ aligned\ pairs\ (p,c)}{total\ number\ of\ c}$
   c. Update p(deletion | c) = $\frac{number\ of\ unaligned\ c}{total\ number\ of\ c}$

Hybrid Phoneme/Grapheme: In the hybrid phoneme/grapheme approach, during recognition each word is modeled with two different phone sequences. The first phone sequence is created manually by native speakers. The second sequence uses a grapheme-as-phone representation. In training, we assume independence between the manual phone set, "*P*" and the grapheme representation, "*G*", and train two different sets of context-dependent HMMs. Words which do not have any manually created pronunciations are spelt with just the grapheme-derived phones. While performing recognition, one can also use pronunciation probabilities to weight the grapheme and phoneme pronunciations differently.

4.2 Experimental results

In the following, we present experimental results on Pashto ASR for comparing the different approaches outlined above. All recognition experiments used a three pass recognition strategy in the BBN Byblos recognizer. The first pass, referred to as the forward pass, uses context-dependent triphones with state-tied mixture (STM) parameter tying and a bigram language model (LM). The second pass, referred to as the backward pass, operates on the lattice from the forward pass using context-dependent quinphones with SCTM configuration for acoustic models and a trigram LM. The output from the backward pass is a lattice or an n-best list. The third and final recognition pass, referred to as the rescoring pass, uses SCTM models trained with crossword quinphones to re-rank the n-best list produced by the backward pass. All acoustic models in the results below were trained using maximum likelihood estimation (MLE). LM training used a total of 700K words from Pashto transcriptions and translations available in the corpus provided by Appen.

Our first experiment was designed to compare the quality of pronunciations produced by standard T2P and the modified version using the improved alignment algorithm. We used a set of 10K manually created word pronunciations to perform the comparison. We compared the two approaches under two operating conditions. In the first condition, we used 1K manually created word pronunciations for training and 9K for testing. In the second, we divided the 10K words equally into two sets of 5K each. Table 2 shows the percentage of words where the predicted pronunciations were identical to the corresponding reference, i.e. the manual pronunciation. From Table 2, we conclude that our updates to the T2P tool outperform the standard tool by a factor of 2 to 3 in prediction accuracy. On analysis of the pronunciation errors from the modified T2P tool, we found that most of the errors are single phone variations in the phonetic string. Therefore, we adopted the improved approach for subsequent experiments that rely on creating automatic phonetic pronunciations.

Train			Test		
#Wds	T2P	Modified T2P	#Wds	T2P	Modified T2P
1K	36%	98%	9K	12%	22%
5K	29%	96%	5K	14%	42%

Table 5. Percentage of words where the predicted pronunciation from the two different text-to-phone are identical to the reference pronunciation

Next, to perform a systematic comparison of different strategies for creation of pronunciation lexicons, we explored three different scenarios by varying the amount of words with manually created phonetic spellings:

1. *Low-resource* that simulates having pronunciation for only the top-1K most frequent words in the training data.}
2. *Medium-resource* with the top-5K words having manual pronunciations.}
3. *Full-resource* where every word has a manual pronunciation.}

For each of the aforementioned scenarios, we trained the following systems:

1. **P**: Phoneme-based systems that are estimated from the corresponding fraction of the audio transcripts for which every word has a manually created pronunciation.}
2. **G**: Single grapheme-based system trained over the entire training set.}
3. **P+G**: Hybrid phoneme/grapheme approach where the recognition dictionary uses two pronunciations (phonetic and graphemic). For words that have a manual pronunciation we use the phonetic representation and for words that do not have a manual pronunciation we use the grapheme representation. During training, we estimate two sets of context-dependent HMMs. The first set uses phonetic representation and is trained from the corresponding fraction of the audio transcripts for which every word has a manually created pronunciation. The second set uses grapheme representation and is trained over the entire available training set. Thus, the grapheme-based HMMs for all three training scenarios are estimated from the same amount of data. This ensures that the grapheme HMMs use all available training data.}
4. **P+T2P**: In this approach, there is a common set of HMMs that use only phonetic representation. For the words that do not have manual pronunciations, the trained letter-to-sound rules from the modified T2P tool are used to create pronunciations automatically. Therefore, the HMMs are trained over the entire training set. The only difference between the three P+T2P systems is the fraction of words with manual and automatic pronunciations.

Table 2 compares the performance of the systems trained from various dictionary configurations as evaluated on the test set in terms of the word error rate (WER). All results are reported with unsupervised constrained maximum likelihood linear regression (CMLLR) speaker adaptation (Gales, 1998). Decoding was performed with the same 10K vocabulary, except for the system P, where the vocabulary size is restricted to the number of words with manual pronunciations. The out-of-vocabulary (OOV) rate for the test set with the 10K vocabulary is 4%, whereas for system P the OOV rate is 5% for the 5K dictionary and 12% for the 1K dictionary.

System	# of Words with Manual Pronunciation			
	0K	1K	5K	10K (all)
P	-	53.2%	46.2%	45.2%
P + T2P	-	45.7%	45.3%	45.2%
P + G	47.3% (G)	46.8%	45.5%	45.1%

Table 6. WER of systems trained from various dictionary configurations as evaluated on the test set. Decoding was based on the same 10K vocabulary, except System P, where the vocabulary is restricted to the number of words with manual pronunciations.

As one would expect, the grapheme system (System G in parentheses in the P+G row of Table 2 results in the worst performance (WER of 47.3%) compared to the systems with the same vocabulary. On the other hand, the phoneme system (System P), which uses manual pronunciations for every word results in a WER of 45.2% - a 2.1% absolute reduction in WER than the grapheme system.

For the low (1K) and medium (5K) resource scenarios the P+T2P and P+G systems yields better performance than the phoneme system. In particular, the P+T2P system significantly outperforms the P+G system for the low-resource scenario (WER 45.7% vs. 46.8%). For the medium-resource scenario, both P+T2P and P+G systems result in comparable performance. Note that the P+G system uses pronunciation probabilities to assign a different weight to the grapheme and phoneme pronunciations.

5. Machine translation

BBN's Statistical Machine Translation (SMT) engine is a phrase-based translation system based on (Koehn, 2004). Word alignments between source-target sentence pairs are generated using GIZA++ (Och and Ney, 2003). In order to improve the quality of the alignments, word alignments in the forward and backward direction are merged as in (Koehn et al., 2003). Phrase pairs are automatically extracted from the word alignments by merging neighboring alignment groups using a set of rules. The decoder uses a log-linear model of different features to choose between competing translation hypotheses. The parameters of the model are estimated using statistics of the phrase pairs extracted from the word alignments. The interpolation weights are optimized by minimizing the translation errors on a held out development set.

The system uses a variety of techniques for increasing accuracy. Among these is the use of multiple alignments, generated from morphological segmentation, as well as a technique for inducing collocations on the English side of the parallel corpus. This technique uses the Minimum Description Length (MDL) principle to find N-grams whose reduction to a single token reduces the overall number of "bits" needed to encode the document. This has the effect of partially "inflecting" the English, so that it better matches an inflected language on the other side of the corpus. Other recent improvements have been the use of phrase alignment confidence (PAC) (Ananthakrishnan et al., 2009) to deal with data sparseness, and context-dependent lexical smoothing for incorporating context.

In this section, we present several enhancements for statistical machine translation in context of speech-to-speech translation. We illustrate these improvements on Pashto/English MT. We first describe our baseline system for Pashto/English translation.

5.1 Baseline Pashto/English MT

Pashto is an inflected language that follows a Subject Object Verb (SOV) word order versus the Subject Verb Object (SVO) word order of English. Nouns and adjectives in Pashto are inflected for gender, number, and case. Verbs in Pashto are complex both in form and in use. Verbs agree in person and number with either the subject or the object of the sentence depending on the tense and the particular construction. One or more affixes can be attached to a word or to each other to form compound words, and components of compound words can be joined or separated depending on style. The different dialects of Pashto show many non-standard grammatical features, some of which are archaisms or descendants of old forms that are discarded by the literary language.

Translation Direction	#Pashto Words		#English Words	
	Total	Unique	Total	Unique
Pashto-to-English (76K sent. Pairs)	1.3M	20.0K	1.1M	10K
English-to-Pashto (34K sent. Pairs)	520K	13.5K	460K	6.7K

Table 7. Description of Pashto/English parallel data available for SMT training.

The data available for training our SMT engine on Pashto/English is shown in Table 3. The amount of data is significantly smaller than the typical broadcast news translation task, where corpora are of the order of several million sentence pairs. Given the fairly small size of the training data, we trained the translation systems on data from both translation directions. The tuning set (held-out from training) comprises of 2K Pashto sentences and 2.3K English sentences. For validation purposes, we report results on a set of 547 sentences for Pashto-to-English (P2E) and 564 sentences for English-to-Pashto (E2P) with 4 references each.

For translating from Pashto-English, we also segment words in Pashto into its constituent "morphemes", that is prefix, stem, and suffix before training in order to improve the quality of the phrase alignments and subsequently the translation. We used the same decomposition algorithm as in (Riesa et al., 2006) to segment our training data. We manually selected 86 prefixes and 68 suffixes in Pashto. Given the list of predefined affixes and uninflected words we iteratively stripped affixes from the word until a valid combination of affixes and stem was found in a large dictionary. Segmentation into morphemes resulted in a 27% reduction in the size of the Pashto vocabulary. It also reduced the number of unknown tokens (untranslated words) by 38% on the validation set.

The training and decoding was performed as follows. We used GIZA++ (Och and Ney, 2003) to generate the word alignments in the source-target and target-source directions according to IBM Model 4. The merged word alignments are used to generate a phrase translation table which contains source-target phrase pairs and associated statistics. The log-linear model includes features computed from the phrase table as well as the target side language model. We use a 4-gram language model trained on 3M Pashto words for E2P and a 5-gram language model trained on 20M English words for P2E. In our experiments, we optimize the feature weights for maximum BLEU on the held-out tuning set. We then decode the validation set with the same configuration but with the tuned weights instead.

5.2 Phrase alignment confidence

In phrase-based statistical machine translation systems, translation performance is contingent on accurate estimation of the translation model parameters derived from the phrase pair statistics. However, data sparsity, an inherent problem in SMT even with large training corpora, often has an adverse impact on the reliability of the extracted phrase translation pairs. A significant proportion of phrase pairs occurs just once (singletons) or a few times in the training data, often resulting in unreliable estimates of the associated statistics. For instance, the unsmoothed estimate of the translation probability of a singleton phrase pair might be very large, but this estimate could be entirely invalid if the pair originated from a word alignment error. Thus, it is desirable to have a measure of phrase pair quality based on the reliability of the underlying word alignments. The lexical smoothing probability used as a feature in the log-linear decoding framework is a well-known, existing measure of phrase pair reliability. In Ananthakrishnan et al. (2009), the notion of alignment entropy as a measure of automatic word alignment quality was used to estimate a probability distribution over the alignments of a given source word, and thus evaluate the uncertainty (entropy) of its Viterbi alignment in the original parallel corpus. Their experiments indicated that alignment entropy is well-correlated with traditional measures of alignment quality, such as Alignment Error Rate (AER). As an extension of alignment entropy, we introduce a feature called phrase alignment confidence as a measure of phrase pair quality derived from an ensemble of parallel corpora obtained by resampling the original training.

We identify occurrences of the same sentence pair in multiple parallel corpora, and determine, based on the corresponding word alignments, whether the phrase pairs extracted from this sentence pair are consistent across the corpora in which it occurs. The technique of bootstrap resampling (Effron, 1979) can be used to construct such corpora. Assuming the parallel training corpus (the pivot) contains N sentence pairs, we create K independent resamples, each of size N, by sampling the original corpus with replacement. On average, about 63% of sentence pairs in each resample will be unique, the remaining being repetitions. Thus, a given sentence pair in the original corpus can be expected to occur 63 of 100 resamples. We invoke the Expectation-Maximization (EM) algorithm to perform automatic word alignment (based on IBM Model 4) on each of the (K + 1) parallel corpora (pivot + K resamples). As each resample contains a different set of sentence pairs drawn from the pivot, the word alignments in each set can potentially be different. During the phrase extraction process, we scan the pivot and identify valid phrase pairs based on the word alignments. When extracting phrase translations from a given pivot sentence pair (S_i,T_i), we identify all resamples R_i, in which that sentence pair occurs, and determine whether the phrase pairs identified in the pivot sentence pair are consistently valid across the resamples. We define the alignment confidence of a single instance of a phrase pair in the pivot as the ratio of the number of resamples in which that instance is identified as a valid phrase pair to the number of resamples in which the containing sentence pair occurs. Note that this measure is computed for each instance of every phrase pair. For non-singleton phrase pairs, we simply take the average of the phrase alignment confidences of each instance across the pivot corpus. Thus, every phrase pair in the pivot corpus now has an associated confidence score in addition to the original statistics. We refer to this measure as phrase alignment confidence.

The discriminative translation framework of the decoder makes it relatively straightforward to add new features to the system. In order to integrate the phrase alignment confidence feature, we simply add to the log linear model an additional term consisting of the new feature and its corresponding weight.

Tables 8 and 9 present results for P2E and E2P SMT after inclusion of the phrase alignment confidence feature in decoding. We resampled the training corpus (pivot) with replacement to generate K=99 resamples for a total of 100 parallel corpora. We then perform augmented phrase pair extraction where, for each instance of every phrase pair in the pivot corpus, we evaluated its consistency across all resamples in which the containing sentence pair occurs. The augmented phrase table encodes this phrase alignment confidence feature in addition to the original statistics. Integrating the proposed phrase alignment confidence feature improved the BLEU score by 3.5% relative on the P2E validation set and 0.4% relative on the E2P set. We believe that while the proposed feature is useful in its own right, it possesses less discriminative power than the standard lexical smoothing feature. The length of a phrase pair does not play a major role in evaluating the phrase alignment confidence feature, whereas longer pairs are almost always de-emphasized by lexical smoothing. In the future, we plan to extend our work on phrase pair quality measurement by taking phrase pair length and the consistency of within-phrase alignments across the resamples into account, making it more competitive with lexical smoothing as well as giving better additive improvements in combination with the latter. We also plan to evaluate the relative usefulness of phrase alignment confidence with respect to the amount of training data available, and to determine whether its importance increases as the training corpus shrinks in size.

Configuration	BLEU	# of Untrans. Words
Baseline	34.8	80
+ Phrase Alignment Confidence	36.0	80
+ Context-Dependent Lexical Smoothing	36.2	80
+ Back-off to a Bilingual Lexicon	36.2	63

Table 8. Experimental Results for Pashto-to-English Text-to-Text Translation.

Configuration	BLEU	# of Untrans. Words
Baseline	24.8	31
+ Phrase Alignment Confidence	24.9	31
+ Context-Dependent Lexical Smoothing	25.1	31
+ Back-off to a Bilingual Lexicon	25.1	16

Table 9. Experimental Results for English-to-Pashto Text-to-Text Translation.

5.3 Context-dependent lexical smoothing

In our phrase-based decoder, the likelihood of translation from a source phrase S = s_1, s_2, ... s_n to a target phrase T = t_1, t_2, ... t_m is primarily modeled with the *rule translation probability* maximum likelihood estimates:

$$P(S|T) = \frac{N(S,T)}{\sum_{S'} N(S',T)}$$

$$P(T|S) = \frac{N(S,T)}{\sum_{T'} N(S,T')}$$

where N(S,T) is the number of times the rule S→T was extracted from the training corpus. However, translation probability is also modeled with an another feature, known as *lexical smoothing* (Koehn, 2004). The forward lexical smoothing score for the rule S→T is defined as:

$$\prod_{i=1}^{m} \sum_{s \in A(t_i|S,T)} \frac{P(t_i|s)}{\|A(t_i|S,T)\|}$$

where $P(t|s) = N(s,t)/\sum_{t'} N(s,t')$ is the probability of the *word-to-word* translation S→T, and *A(t|S,T)* is the set of source words aligned to *t* in the rule S→T. In this case, N(t,s) counts the number of times *s* is aligned to *t* in the GIZA aligned training data. Also note that either *s* or *t* can be NULL.

The backwards lexical smoothing score is analogously:

$$\prod_{i=1}^{m} \sum_{t \in A(s_i|S,T)} \frac{P(s_i|t)}{\|A(s_i|S,T)\|}$$

Note that the lexical smoothing score is computed at the word level without factoring in local context, even though intuitively we know that context is important for both human and machine translation accuracy. On the other hand, the average word-to-word translation will be seen far more times in the training than the average phrase-to-phrase translation, so the word-level maximum likelihood estimates P(t|s) and P(s|t) will be estimated than the phrase-level maximum likelihood estimates P(S|T) and P(T|S).

Ideally, we would like to harness the increased contextualization of the rule translation probabilities without sacrificing the accuracy of the word-to-word maximum likelihood estimates. To that end, in the *context-dependent lexical smoothing* approach we condition the word translation probabilities on local context, and then interpolate them context-independent probabilities to ensure that the final probabilities are well-estimated.

We currently use *previous word* and *next word* as context types. Formally, these context-dependent lexical probabilities are represented as:

$$P(t|s_i, s_{i-1}) = \frac{N(s_i, s_{i-1}, t)}{\sum_{t'} N(s, s_{i-1}, t')}$$

$$P(t|s_i, s_{i+1}) = \frac{N(s_i, s_{i+1}, t)}{\sum_{t'} N(s, s_{i+1}, t')}$$

Rather than directly interpolating the probabilities or using an explicit back-off model, we simply interpolate the lexical counts:

$$P(t|s_i) = \frac{N(s_i, t) + \alpha N(s, s_{i-1}, t) + \beta N(s, s_{i+1}, t)}{\sum_{t'} N(s_i, t') + \sum_{t'} \alpha N(s, s_{i-1}, t') + \sum_{t'} \beta N(s, s_{i+1}, t')}$$

where C(s$_i$) is the local context of source word s$_i$, and the interpolation weights α and β are globally optimized on a tuning set. This type of count-based interpolation acts as an implicit ``back-off'' model, since the more times a particular context type has been seen, the more mass it adds to the final probability.

The interpolated probability P(t|s,C(s)) is used in the standard lexical smoothing formula and this score is used as an additional log-linear decoding feature. We also use context-dependent lexical smoothing in the backwards direction, conditioning on target context.

Tables 8 and 9 summarize the impact of using context-dependent lexical smoothing for P2E and E2P SMT. As shown in the two tables, there is a modest improvement in BLEU scores in both directions.

5.4 Effective use of bilingual lexicon

Often, the heuristics used to determine valid phrases in the phrase extraction step result in unaligned source-target words occurring in the corpora being omitted from the phrase translation table. Hence, a word that appears in the training corpus is not guaranteed to have a translation during decoding. The use of a supplementary bilingual translation lexicon that covers such words improves the coverage of the system. Traditionally bilingual translation lexicons are used as additional training data for machine translation systems and allowed to drive the word alignments. However, the entries in a lexicon have such high phrase translation and lexical probabilities that they can cause serious word sense errors if the particular source word occurs in a different context. If a word that occurs in the lexicon is identified in the input sentence, its corresponding single word translation from the lexicon will almost always be preferred over a longer phrase pair whose source phrase contains that word. We tackle this issue by backing off to entries in the lexicon only if the source word cannot be translated as part of a source phrase existing in the phrase translation table.

Using a bilingual expert, we created a bilingual lexicon consisting of a total of 30K entries. In our experiments, using the bilingual translation lexicon did not improve the BLEU metric, however it resulted in a 50% reduction in untranslated words for E2P and 21% reduction for P2E as shown in Tables 8 and 9.

6. Evaluation

From 2006 to 2010, BBN TransTalk has been evaluated in several US Goverment sponsored evaluations conduced by an independent third party such as NIST and MITRE. In these evaluations participating S2S systems are evaluated on several dimensions include rate of concept transfer in live interactions with role players, odds of concept transfer on offline data, and automated metrics such as word error rate (WER) and BLEU scores computed on offline recorded audio. User surveys based on questionnaires are also used to measure the ease of use, efficacy of interaction, etc. based on users' impression of the live interaction.

Table 10 summarizes BBN's performance as measured against the following official program metrics in the program for the past three years. The evaluations were typically on a different language and often on different platforms.

High-level Concept Transfer (HCT): This metric is computed from live interaction of users with the system in an allotted time interval (typically 20 minutes). A team of bilingual judges compares the output of the TRANSTAC system to what was spoken by the role-playing US military personnel, i.e., subject matter experts (SMEs) and foreign language speaker (FLS). The judges are asked to rate, on an utterance-by-utterance basis, how well the utterance spoken by the human speaker was translated by the system and how many times the speaker attempted the utterance. When multiple attempts were made, only the best translation was scored. Both English to foreign language and foreign language to English directions were scored. The translation quality has four possible scorings:

1. Unknown – The utterance in the scenario was not attempted by the SME or FLS. A score of "0" is assigned to this category.
2. Inadequate – None of the concepts came across in the utterances.A score of "0" is assigned to this category.
3. Partially adequate – Some of the concepts came across in the utterance.
4. Adequate – All of the concepts came across in the utterance.

Partially adequate are given a score of 0.5, and adequate are given a score of 1. In the case where multiple concepts were provided by the FLE in response to the SME's question, each answer is counted separately. These scores are then aggregated over the entire session, and the transfer rate per ten minutes of conversation is computed.

Odds of Successful Low-Level Concept Transfer (LLCT): This metric is computed using the system output and reference translations on an offline, pre-recorded data set. A bilingual human annotator identifies low-level concepts (such as "car", "door", "black color", etc.) that are correct or incorrectly transferred in the system output. Next, the odds of successful transfer of these low-level concepts are computed by dividing the number of successes by the number of errors. The higher the odds of success, the better the system.

Subject Matter Expert (SME) Utility Assessment (SUA): This metric is computed from responses to questionnaire by SMEs after interacting with the system in any given session. A utility score is computed by aggregating scores across sessions for each type of question. The questions in the SME questionnaire range from: "I found the system easy to understand in this interaction" to "I would use this system in the field in its current state of functionality".

Table 10 describes the performance of BBN systems in the evaluations in reverse chronological order with most recent evaluations at the top of the table.

Eval. Date	Language	Platform	HCT		LLCT		SUA
			E2F	F2E	E2F	F2E	
Aug 2010	Dari	Smartphone	15 (1st)	25 (1st)	3.3 (2nd)	1.5 (1st)	1st
April 2010	Pashto	Smartphone	19 (1st)	30 (1st)	4.2 (1st)	3.0 (1st)	1st
June 2009	Dari	UMPC	13 (1st)	14 (1st)	3.5 (1st)	1.6 (1st)	1st
Nov 2008	Iraqi	Laptop	22 (2nd)	28 (2nd)	7.3 (1st)	5.6 (1st)	1st

Table 10. Performance of BBN TransTalk in recent DARPA TRANSTAC evaluations.

7. Conclusions

We have presented our speech-to-speech translation system, TransTalk, and outlined several techniques for overcoming challenges in languages that it has been configured in. For ASR, we described an approach for configuring the ASR system with limited amount of manual pronunciations. Our approach extends existing approaches for languages that have significant mismatch in number of phonemes and graphemes, and shows that comparable performance to a full lexicon can be achieved by creating manual pronunciations for a small fraction of words in the vocabulary. For MT, we discussed techniques for overcoming challenges due to data sparsity such as the use of phrase alignment confidence and effective backoff to a bilingual dictionary. We also presented a method for evaluating the effectiveness of different user confirmation strategies, and shown that back-translation provides higher precision than the simple strategy of reading back the ASR, at the expense of recall.

8. References

Akbacak, M., Franco, H., Frandsen, M., Hasan, S., Jameel, H., Kathol, A., Khadivi, S., Lei, X., Mandal, A., Mansour, S., Precoda, K., Richey, C., Vergyri, D., Wang, W., Yang, M., Zheng, J., 2009. Recent advances in sri's iraqcomm: Iraqi arabic-english speech-to-speech translation system. In: ICASSP '09: Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing.

Ananthakrishnan, S., Prasad, R., Natarajan, P., 2009.Alignment entropy as an automated measure of bitext fidelity for statistical machine translation.In: Proceedings of the 7th International Conference on Natural Language Processing.

Belvin, R., Ettelaie, E., Gandhe, S., Georgiou, P., Knight, K., Marcu, D., Narayanan, S., Traum, D., 2005. Transonics: A practical speech-to-speech translator for english-farsi medical dialogues. In: Proceedings of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING). pp. 89-92.

Bikel, D. M., Schwartz, R., Weischedel, R. M., 1999.An algorithm that learns what's in a name.In: Machine Learning Special Issue on Natural Language Learning.

Billa, J., Noamany, M., Srivastava, A., Liu, D., Stone, R., Xu, J., Makhoul, J., Kubala, F., 2002. Audio indexing of arabic broadcast news. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing.Vol. 1, pp. I-5 – I-8.

Black, A., Lenzo, K., Pagel, V., 1998. Issues in building general letter to sound rules. In: ESCA Workshop on Speech Synthesis. pp. 77-80.

Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, A., Sanchis, A., Ue±ng, N., 2004.Con¯dence estimation for machine translation.In: COLING '04: Proceedings of the 20th international conference on Computational Linguistics. Association for Computational Linguistics, Morristown, NJ, USA, p. 315.

Efron, B., 1979. Bootstrap methods: Another look at the jackknife. The Annals of Statistics 7 (1), pp. 1-26.

Gales, M., 1998.Maximum likelihood linear transformations for hmm-based speech recognition.Computer Speech and Language 12 (2), 75-98.

Gao, Y., Gu, L., Zhou, B., Sarikaya, R., Afify, M., Kuo, H.-k., Zhu, W.-z., Deng, Y., Prosser, C., Zhang, W., Besacier, L., 2006. Ibm mastor system: Multilingual automatic speech-to-speech translator. In: Proceedings of HLT Medical Speech Translation Workshop.

Koehn, P., 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In: AMTA. pp. 115-124.

Koehn, P., Och, F. J., Marcu, D., 2003.Statistical phrase-based translation. In: NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. Association for Computational Linguistics, Morristown, NJ, USA, pp. 48-54.

Magimai-Doss, M., Bengio, S., Bourlard, H., may. 2004. Joint decoding for phoneme-grapheme continuous speech recognition. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing. Vol. 1.

McCallum, A., Nigam, K., 1998. A comparison of event models for naïve bayes text classi¯cation. In: Proceedings of AAAI-98 Workshop on Learning for Text Categorization. AAAI Press, pp. 41-48.

Nguyen, L., Schwartz, R., 1997.Efficient 2-pass n-best decoder. In: DARPA Speech Recognition Workshop. pp. 167-170.

Och, F. J., Ney, H., March 2003.A systematic comparison of various statistical alignment models. Computational Linguistics 29 (1), 19-51.

Prasad, R., Moran, C., Choi, F., Meermeier, R., Saleem, S., C., K., Stallard,D., Natarajan, P., 2008. Name aware speech-to-speech translation for english/iraqi. pp. 249-252.

Prasad, R., Tsakalidis, S., Bulyko, I., Kao, C.-l., Natarajan, P., 2010.Pashto speech recognition with limited pronunciation lexicon. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing. pp. 5086-5089.

Riesa, J., Mohit, B., Knight, K., Marcu, D., 2006. Building an english-iraqi arabic machine translation system for spoken utterances with limited resources. In: Proceedings of ISCA Interspeech.

Stallard, D., Choi, F., Kao, C.-L., Krstovski, K., Natarajan, P., Prasad, R., Saleem, S., Subramanian, K., 2007.The bbn 2007 displayless English/Iraqi speech-to-speech translation system.In: Proceedings of ISCA INTERSPEECH. ISCA, pp. 2817-2820.

Part 2

Language Learning

4

Automatic Feedback for L2 Prosody Learning

Anne Bonneau and Vincent Colotte
LORIA/CNRS, LORIA/UHP (Université Henri Poincaré), France

1. Introduction

The emergence of speech signal processing functions has allowed speech scientists to analyse and modify speech with the aim of improving the perception and/or the production of speech in adverse conditions and language learning. In most cases, these tools are all the more efficient that they take into account the phonological system of each language.

The applications of these aids are numerous and include the improvement of speech intelligibility in noise (Hazan & Simpson, 1998), and for hearing aids (Loizou, 1998), computer assisted-aids for language learning devoted to speech therapists or learners of a foreign language. They are concerned either with speech intelligibility (comprehension), or with the perception and production of speech sounds and prosody. One of the best known works in this domain, the study by Tallal (Tallal et al., 1996), published in Science, proposed speech modifications for hearing impaired children.

Among signal modifications, the enhancement and slowing down of specific regions of speech signals has been the object of numerous studies and applications. Ortega & Hazan (1999) applied these techniques to the improvement of speech intelligibility in second language learning. Colotte et al. (2001) also tackle speech intelligibility in second language learning. By means of an entirely automatic method, they enhanced unvoiced stops and fricatives and slowed down transitions.

This paper deals with the elaboration of advanced feedback devoted to aid learners in the acquisition of the prosody of a foreign language, and presents a pilot experiment investigating the immediate impact of such feedback on learners. Note that, from now on, we will use L2 for second (non-native) language and L1 for the learners' first (native) language. A computer-assisted aid in language learning (a CALL system) and more precisely in prosody can offer at least three kinds of feedback: (1) visual feedback, such as visual displays of the learners' melodic curves, often associated with those of reference speakers (2) automatic diagnoses, based upon acoustical analyses of learners' realisations and (3) "advanced" perceptual feedback, through speech manipulations.

Visualisation of melodic curves has been proposed since the early 60s, not only in second language learning but also in other domains such as hearing deficiencies. Vardanian (1964) was one of the first scientists to use melodic curve visualisation in second language learning and test its impact on learners (Brasilian students learning English). Her results, may be due to the poor quality of visualisation (oscilloscope displays) were far from convincing. Nevertheless, after a first period of scepticism, speech specialists, like James (1977), who used Philippe Martin's melodic curve detection (Germain-Rutherford & Martin, 2000), or De Bot (1983), agreed on the efficacy of visual patterns. Current commercial computer-assisted aid in language learning systems, such as "Tell me more" of Auralog, LangMaster or Better Accent (Kommissarchik & Kommissarchik, 2004) propose visual display of learners' melodic curves.

The simple visualisation of melodic curves, although interesting, is not sufficient if one wants to provide learners with clear feedback about their production. Indeed, as Chun (1998) noted, if no further feedback is provided, the learners have to "extrapolate" their deviations themselves. A more efficient aid would provide learners with indications about their deviations and the way to improve their realisations. But the elaboration of automatic diagnoses encounters at least two major problems. Firstly, an automatic diagnosis of a learner's intonation presupposes the segmentation and labelling of the signal in syllables and in speech sounds. Yet this automatic segmentation, made by alignment if the text is known, is risky for non-native speech. (See 2.1.). A second major problem stems from the notion of "error" or "deviation". Indeed, we face issues such as "what are the links between acoustic cues and categories?" and "what are the accepted deviations?" For the moment the way to deal with this problem consists in proposing an evaluation of the degree of difference between prosodic patterns produced by a reference (or references) and those of the learner. Impact of feedback (diagnosis) on English intonation for French advanced learners (students in English at university) has been elaborated and tested by Herry and Hirst (Herry & Hirst, 2010). Learners did not improve their realisations due to diagnosis. The authors underlined that students participating in the test were volunteers, which might have influence results.

Another interesting way of helping learners in language learning relies upon speech modifications. Winpitch LPL (Martin, 2004), a speech signal editor, proposes functions especially designed for L2 teachers and learners, which enable the user to modify by hand fundamental frequency[1] and duration and annotate prosodic displays. Manipulations of prosodic cues are intended to make learners aware of prosodic patterns that do not exist in their first language; they are in general realized through PSOLA (Pitch Synchronous Over-Lap and Add) resynthesis. An interesting exploitation of speech manipulation consists in replacing the learner's prosodic cues by those of a reference without modifying the learner's timbre. WinSnoori (Laprie, 1999), software for speech analysis, enables users to realize this substitution by hand. An automatic version of this substitution has been realized (Henry et al., 2007) and implemented in WinSnoori. Other speech transformations based on an accent morphing technique have been recently proposed in the domain of foreign language intelligibility (Ingram et al., 2009; Yanagisawa & Huckvale, 2007). In particular, the last authors modified the accent of the speaker whilst maintaining his identity to improve the intelligibility of foreign-accented speech.

In this paper, we will present the various kinds of feedback (visual displays, diagnosis, and perceptual feedback based upon prosodic cues substitution)[2] we elaborate for L2 prosody learning (section 2), and a pilot experiment devoted to test their immediate impact (section 3). Results are given in section 4.

2. The L2 prosody learning platform

A platform designed for manual and automatic feedback of learners' prosody has been implemented in a research version of WinSnoori software. The platform contains classical

[1] The variations of the fundamental frequency (F0) generate the melodic curve.

[2] Parts of these tools have been presented in (Henry et al. 2007)

functions such as spectrographic displays, speech recording, real-time F0 (fundamental frequency) display, and integrates three particularly interesting functions with respect to language learning: an automatic text-to-speech alignment for native and non-native speech, a module for modifying the speech signal manually (the fundamental frequency and duration can be modified at the same time), and another module displaying automatic feedback on learners' (non-native) productions and exploiting most of the above-mentioned functions.

Besides classical F0 displays, two kinds of feedback are provided to learners, each of them based upon a comparison between a reference and the learner's production. The first feedback, a diagnosis, provided both in the form of a short text and visual displays such as arrows, comes from an acoustic evaluation of the learner's realisation; it deals with two prosodic cues: the melodic curve, and phoneme duration. The second feedback is perceptual and consists in a replacement of the learner's prosodic cues (duration and F0) by those of the reference. We will first describe the phonetic alignment, necessary for automatic diagnoses and speech modifications, then the method developed to modify speech signals, and, finally, perceptual feedback, diagnosis and the modus operandi.

2.1 Automatic text-to-speech alignment

Prosodic cues generally appear on well determined linguistic and phonetic entities. So a preliminary segmentation into words, phones and syllables is necessary to localize the prosodic events and to compare the learner's realization with that of a reference. After users produce a linguistic entity (a word, a group of words, or a sentence) from the corpus, a segmentation of their realization is performed. First, a phonetization of the text is carried out using the CMU dictionary. Then, the segmentation is computed with a text-to-speech alignment, which establishes the correspondence between phonetic units and parts of the speech signal. Text-to-speech alignment is achieved using Hidden Markov Models (Fohr et al., 1996).

Two different kinds of model have to be used: one for native speakers and another one for non-native speakers (learners). Indeed, since learners of a foreign language tend to replace the sounds they do not know by sounds of their first language, models used for native speakers (learned on the TIMIT database and developed for automatic speech recognition - ASR- purposes) should be adapted to non-native speakers. Although we have already at our disposal an ASR system designed for non-native speech recognition (Bouselmi et al., 2005) we are still working on non-native alignment. Indeed, if ASR systems do not need precise segment boundaries for typical applications such as automatic speech transcription or translation, this is not the case for language learning applications where the production of a subject is analysed and corrected. The need for precise boundaries is obvious for diagnosis about segment durations, but is also important for speech modification functions such as the ones we design. Since the detection of very precise segment boundaries is not always possible, we need to associate confidence levels with detections. Then feedback can be avoided in the cases where detections are not sufficiently reliable. We are presently working on these two aspects of ASR for non-native speech: the improvement of segment boundary precision and the elaboration of confidence levels. The syllabification program of NIST was applied to the CMU dictionary in order to obtain a database of syllabified words.

2.2 Speech signal modification algorithm

Signal modification functions have been included in the platform. These functions are based on an improved version of well known TD-PSOLA method (Colotte & Laprie, 2002;

Moulines & Charpentier, 1990) and allow users to manually modify F0 contours, speech rates as well as phoneme durations. It means that we can apply a global or local modification factor. For instance, we can slow down a particular part of the signal and speed up another one of the same signal.

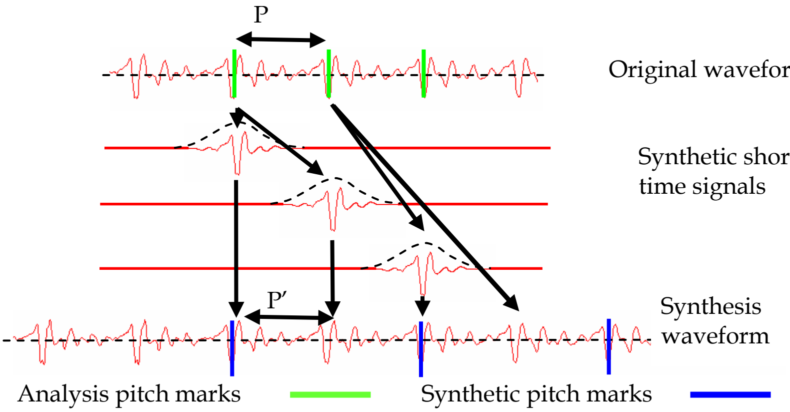


Fig. 1. Example of slowing down: the duration of the original signal is lengthened by a factor 2 (P'=P).

The modification method is based on the decomposition of the speech signal into overlapping pitch synchronous frames and the modification of pitch or duration is obtained by duplication/decimation of some frames.

Firstly, this method supposes the detection of the pitch marks: the signal periods are marked at the maximum or minimum relevant peaks (for the voiced parts[3]). These marks are spaced every pitch period and indicate the centre of the (analysis) frames (see fig.1 and 2).

Secondly, we need to compute the new position of these marked frames in the modified signal. The main requirement is to maintain the consistency of mark location between frames in order to preserve the original temporal structure of the signal under analysis. This marking directly influences the quality of the resulting signal. In (Colotte & Laprie, 2002), we have proposed a high precision algorithm for pitch marking at two levels: analysis and synthesis marks. In one hand, dynamic programming selects peaks in the signal for marking periods. Through correlation and pruning strategies, the algorithm overcomes errors which may appear with other algorithms. In addition, the algorithm is very fast in computation, which is very suitable for TD-PSOLA method. In the other hand, the combination of our pitch marking with a fast re-sampling method (to obtain the true synthesis frame) during the synthesis step increases the signal quality. This gain in accuracy avoids the reduction of quality between original and synthetic signal observed with the classical TD-PSOLA method: the level of noise between harmonics is reduced with our method.

Thirdly, each mark in the future signal is associated with a mark of the original signal. If we want to slow down (resp. speed up) the signal, the space between frames needs to be preserved (to keep the same pitch) and an original frame could be duplicated (resp. removed) to obtain the good length of the signal (see fig.1). If we want to modify the pitch, preserved, as for the slowing down, an original frame could be duplicated (or removed) to

[3] For unvoiced part, by definition without period, an arbitrary spacing is used (for instance 5ms).

obtain the good length with the good spacing (see fig.2). The pitch and duration modifications can be merged together.

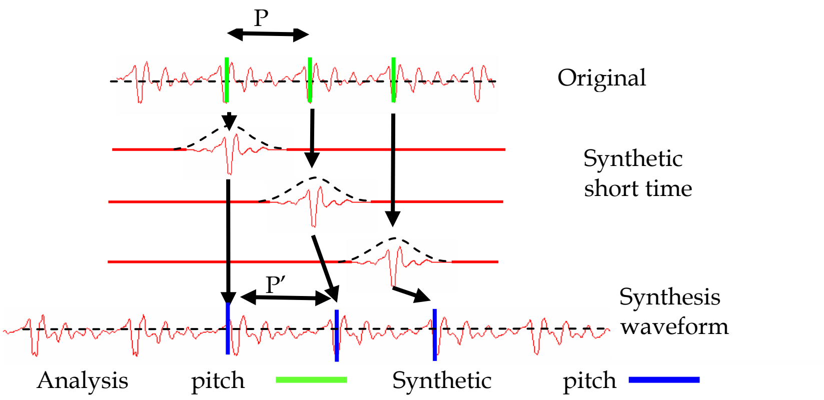


Fig. 2. Example of pitch modification (here a lengthening of the period/a lowering of the F0): the original pitch is lengthened by a factor 1.2 (P'=1.2P).

Finally, the signal is rebuilt from these frames (removed and/or duplicated) thanks to the principle of overlapping.

The advantages of this method are that we work in the temporal domain without any computation of transformation into the frequency space and the principle of using frames synchronized with the pitch does not destroy the coherence of the signal for each frame (notion of shape-invariance). We obtain a high quality of resynthesis without modification of the timbre of the voice. The method allows us to lengthen the pitch period until 3 times the initial period (i.e the F0 can be lowered by 3). For the duration, the slow down is only limited by the fact that a numerous duplication of the same part can create a new period (for unvoiced sound) and a noise of voicing can appear where there was no voicing (the factor > 4-5). But the slowdown is sufficiently local and generally small enough to avoid this drawback.

These modification functions can be use independently of the diagnosis process. The learner or teacher can (manually) modify the signal as he wants, to become aware of the link between prosodic cue variations and perception.

2.3 Automatic perceptual feedback

Such functions can be exploited to imitate the prosodic cues of a model, so that learners can appreciate the differences between their realization and what they are expected to realize. The signal is resynthesized with the required modifications, and the users can listen to the modified signals as well as visualize their spectrographic representations and the new melodic curves.

We have developed a module which realizes this imitation automatically. In a first step, the relative durations of the learner's phones are aligned with that of the reference. In a second step, a new F0 contour for the learner's utterance is computed using a linear interpolation of the model's normalized F0 contour. Then the learner's realization is resynthesized and then he/she can appreciate the resulting speech signal.

Note that the used resynthesis algorithm keeps the timbre of the learner's voice, since only F0 and phoneme durations are modified. The copy of the F0 takes into account the mean F0 (pitch) of the speaker.

2.4 Diagnosis

The diagnosis is based upon a comparison between the realisation of the learner and that of the reference speaker, as well as phonetic knowledge about prosodic patterns of L1 and L2. Let us take the example of the realisation of English lexical stress in isolated words by French speakers (the object of the experiment presented in section 3). The syllable which should be stressed is assumed to be the one exhibiting the higher F0 in the reference's production. Thus the system evaluates, in semi-tones, the peak height of this syllable with respect to other syllables, in both realisations (the native and non-native ones) and returns a comment indicating whether the prominent F0 peak appears on the expected syllable of the non-native realisation and if it is sufficiently marked. At the same time, visual displays are shown on the spectrogram of the learners' realizations: arrows indicate whether the pitch of the target syllables should be raised or lowered (the colour of each arrow provides an indication about the degree of difference between native and non-native realisations with respect to F0 height); a red curve represents the F0 contour; the syllable and vowel durations of the reference and those of the learner appear in the form of bars on the top of the learner's realization; the length of bars varying with the duration of these segments (see Fig. 3). The reduction phenomenon requires a specific treatment. If learners do not reduce syllables which have been strongly reduced by the English reference, they are invited to repeat their realization and to reduce the appropriate syllable.

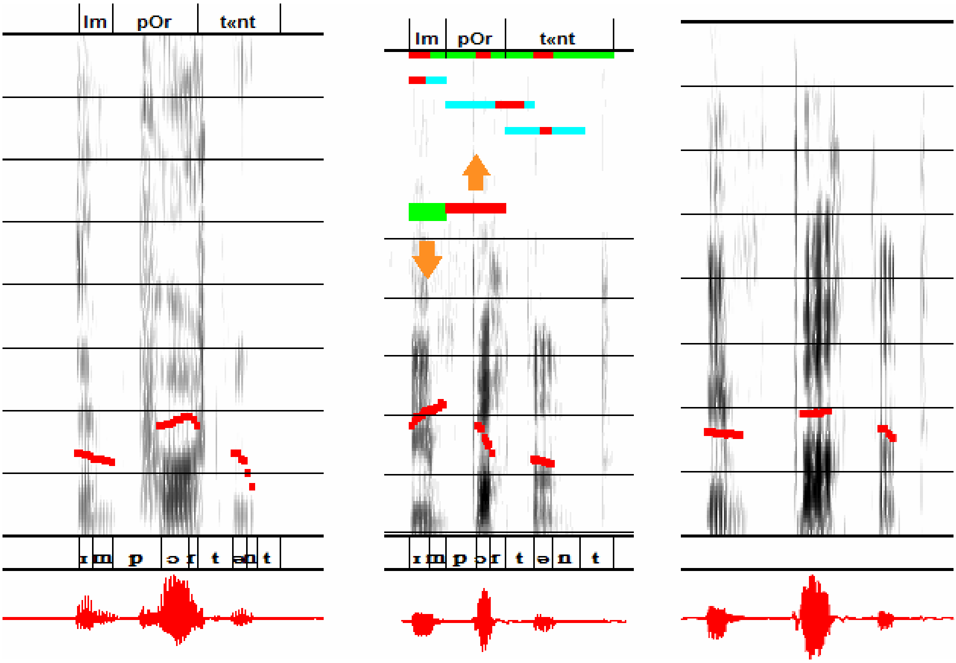


Fig. 3. Example of automatic diagnosis and speech modifications. Each panel shows the spectrogram, the melodic curve (in red) and the waveform (at the bottom, also in red) of a realization of the word "important". The first panel show the realization of this word by an English speaker; the second panel its realization by a French learner, with the diagnosis provided by the software (see text), and the third panel the modification of the learner's voice.

2.5 Modus operandi

Let us present the procedure proposed for providing automatic feedback to learners of a foreign language. Before clicking on the function in the menu, the subject should record a word or a small sentence, belonging to our database. He can, depending upon his wish or the task he is involved in, listen to the English reference before producing the item. When the subject selects the "correction" function, he is asked to choose a reference (one male and one female English speakers have recorded the corpus). Then the software realizes a text-to-speech automatic alignment of the native and non-native versions, and substitutes the prosodic cues of the native speakers to those of the non-native speakers.

Then the subject is provided with three versions of the utterance analysed, displayed in three windows, one per each version (see fig. 3). The first version consists in the native speaker's production, the second version is the learner's production, and the third version is the learner's modified production. Each window contains the representation of the melodic curve onto the spectrographic display, with the automatic alignment just under the spectrogram. The automatic diagnoses appear onto the spectrogram of the second window, that of the learner's original production. The subject can listen to the three versions and is invited to give a special attention to his own modified voice. The subject is free to produce the word again or select another one.

3. Experiment

We conducted a pilot experiment to analyse the immediate impact of diagnosis and advanced perceptual feedback on French subjects speaking English as a second language. In France, teaching of English prosody for nonspecialists focuses on the main intonation patterns as well as the place and strength of English lexical stress accent. We have chosen to test the production of English lexical accent in isolated words by French speakers. French and English are very different from a prosodic point of view since French is considered as a syllable-timed language and English as stress-timed language (Dauer, 1983). The place of the English lexical accent is free, whereas the French one is fixed, and the English accent is very well marked on an acoustical point of view whereas the French one is relatively weakly marked. Indeed, in English, the stressed syllable of isolated words is more intense, higher in pitch and longer than the unstressed ones. Furthermore, post-stressed syllables are often reduced: their vocalic nucleus tends to be very short, weak, and its vocalic timbre can become similar to that of a schwa (the schwa is, as an example, the first and last vowels in the word "America"). The French accent is essentially characterized by a lengthening of the last syllable of a word or a group of words; non accentuated vowels are always produced with their "full" timbre (there is no reduction).

It is commonly said that learners of a foreign language are "deaf" to its prosodic system. This explains why learners tend to use the prosodic features of their mother language instead of the prosodic features of the target language. For example, when speaking English, French learners generally lengthen the last syllable of a word, even when they know that this syllable is unstressed, and produce post-stressed vowels with their full timbre (without any reduction).

3.1 Experimental protocol

3.1.1 Corpus and subjects

The corpus, made up of 40 English transparent words, has been recorded by two English speakers, the "reference" speakers, one male and one female, born and educated in England,

and currently living in France. Transparent words have similar spelling in French and English but do not have the same pronunciation (e.g. "important", "favourite"). They constitute a good way of making learners aware of the differences between accentuation in both languages. The corpus was made up of two and three-syllable words of the type: `S1S2 (the accent is on the first syllable), S1`S2S3 (accent on the second syllable) and `S1(S2)S3. Words with this last syllabic structure (e.g. "governor", "family"), are generally pronounced by English speakers with a very strong degree of reduction on the second vowel whereas French speakers pronounce this vowel with its full timbre. We kept ten words for a "familiarization" phase, which precedes all the recording sessions; these words were pronounced by speakers but not analysed. Thus thirty words –ten per each type- remained for the analysis. Note that, in this corpus, the lexical accent never appears on the last syllable of a word.

Ten French subjects, five male and five female speakers from 15 to 50 years old, participated in the experiment. Taking into account learners' pronunciation, we chose four (relatively) advanced speakers and six low (production) level speakers. The advanced speakers have lived for a short period in USA or made frequent trips in this country, whereas the six speakers with a low oral production level (let us call them beginners, for short) have studied English at school for at least five years but do not master English pronunciation. Since ten subjects produced thirty words in each of the two recording sessions (presented below), the corpus contains 600 words uttered by non-native speakers.

English reference speakers and French subjects recorded the corpus in a quiet office with a Sennheiser headset microphone connected to a laptop. The mono signal was digitized at 16 bits with a sampling frequency at 22500 Hz. The software Audacity was used for all the recording sessions except for the one requiring the use of the platform (second session of the test condition).

3.1.2 Experimental conditions

There were two experimental conditions: the test and control ones, and two sessions of recording for each condition. We selected five speakers for each condition, two advanced speakers and three "beginners", trying to balance both groups. Note that we do not let learners chose between the test or the control conditions.

The first session was the same for both conditions, subjects from both groups (the test and control groups) recorded the corpus without listening previously to English references. This gives us an indication of the learners' mastering of English lexical accent at the beginning of the experiment. Each word was written on a sheet of paper in orthographic transcription. To avoid the production of a list intonation, a short pause was enforced between each word.

In the second session, subjects participating in the test condition were submitted to the platform for prosody learning. The procedure was the following: subjects were invited to select a word, written in orthographic transcription, from a list corresponding to the corpus. For the purpose of this experiment we asked subjects to follow the list order, and select a male or a female reference according to their gender. Then subjects uttered the selected word without listening previously to its English reference; the system analysed the realisation, and visual, perceptual and textual (diagnosis) feedback was displayed on three windows as explained in section 2.5. After subjects took knowledge of the various kinds of feedback, they recorded the word one more time, and selected the following word to repeat the procedure until the end of the list. Only word repetitions (the pronunciation after feedback) have been analysed and compared to words pronounced during the first session.

In order to avoid false corrections due to erroneous segment boundary detection, the system stops just after the automatic alignment of the word under analysis to ask the experimenter whether he/she desires to continue the process or modify speech boundaries before. This is a necessary step if we want to test the impact of feedback without incurring the risk of disconcerting the learner by inappropriate corrections. Let us recall that we are currently working on the elaboration of confidence levels to limit the risk of erroneous corrections.

In the second session, subjects participating in the control condition read each word on a sheet of paper, listened to its reference, and uttered it.

Both conditions enable us to compare the effect of simple auditory feedback, received by learners in control condition, with that of advanced feedback, received by learners in test condition.

3.2 Acoustic cues

We estimated differences in height (F0) and duration ratios for `S1S2 and S1`S2S3 words with the lexical accent falling on the penultimate vowel, and we analysed the presence or absence of reduction in the second syllable of `S1S2S3 words.

To analyse all productions, including those not analysed by the platform during the test condition (second session), we segmented words into speech sounds, and estimated the values of the acoustic cues taken into account, i.e. fundamental frequency and segment duration. F0 was evaluated in semi-tones.

For words with `S1S2 and S1`S2S3 syllabic structures, we estimated the following criteria.

1. The differences between F0 values of the (to be) stressed vowel (VS) and the following unstressed vowel (VUF, for unstressed final vowel) :
   F0(VS) - F0(VUF) .
   F0 was averaged across all the frames of each vowel except the frames close to the vowel boundaries.
2. The number of times the F0 maximum fell on the right syllable (the theoretically stressed one), given in percentage.
3. The ratio of the duration of the stressed vowel to that of the following unstressed vowel:
   D(VS)/D(VUF).
   Note that VUF is the last vowel of the word, and could be lengthened by French subjects. Taking into consideration ratio and not difference in the above formula allows us to remove the effect of temporal variations due to speech tempo.

For `S1(S2)S3 words, we noted whether French learners pronounced the second vowel (V2) with its full timbre. To estimate the presence/absence of reduction, we calculated the duration of V2, when it was audible, and we compared it to the averaged duration of V1 and V3. If the duration of V2 was inferior to 40 ms and at least two times shorter than the averaged duration of other vowels in the word, we considered that the vowel had been reduced. With these criteria, all the second vowels in `S1(S2)S3 words uttered by the English speakers used as references in this study are considered as reduced.

We had also compared F0 height and durations between the stressed and the initial unstressed vowels (for S1`S2S3 words) but these parameters did not contribute a lot to global results and we do not give them here in order to simplify the discussion.

We used Student’s t-tests, paired samples, associating the data of each speaker and each word to compare the pronunciation of the words in the first session (without feedback) to their pronunciation in the second session (with feedback). Each condition (auditory feedback and advanced feedback) and each group of speakers (i.e. advanced speakers and beginners) were tested separately. We also used Student's t-tests to compare results obtained in both conditions (only words uttered during the second session were taken into account). Once more advanced speakers and beginners were considered separately. We accepted a level of 0.05 for significant effect and considered that results were highly significant when this level was inferior to 0.001. We submit the following parameters to statistical tests: the first parameter (F0 height), the third parameter (relative duration ratio) as well as, for vowel reduction in 'S1S2S3 words, the ratio of the duration of the second vowel to that of other vowels in the words.

4. Results

Table 1 provides, for each session, the results obtained for advanced learners and beginners as well as the results for both reference speakers. Parameter values are averaged across all words and all speakers of a given level (advanced or beginner). For estimating the averaged differences in height (in semi-tones) between stressed and unstressed vowels (column 2), we only took into account the cases where the F0 maximum is on the correct syllable. The third column displays the number of times the F0 peak is located on the right syllable, and the fifth column the number of times the second vowel in `S1S2S3 words is reduced. The duration ratios are given in the fourth column.

We also calculated the averaged values obtained for each speaker in each parameter, and we discuss them below, when interesting.

Speakers	F0 (ST)	Max F0 place (%)	Duration ratio	Reduction (%)
REF (M)	11	100	1.7	100
REF (F)	9	100	1.7	100
Adv.	6.9	90	1.05	7.5
Adv.	7.4	100	1.2	100
Adv.	7.5	100	1.25	100
Beg.	2.5	33	0.85	2
Beg.	2.9	43	0.83	37
Beg.	3.9	75	1.1	63

Table 1. Results for the first session, without feedback (blue line) and for control and test conditions with feedback (white and green lines, respectively), for advanced learners (Adv), and beginners (Beg). Values for reference speakers, the male (M) and female (F) are given on the top of the table. See text for more explanations.

4.1 First session

In this session, all learners (from test and control groups) uttered the words of the corpus without listening previously to their English references.

4.1.1 Reduction phenomon in `S1S2S3 words

Subjects utter these words the French way: they realize post-stressed vowels with their full timbre. We note only few exceptions: one advanced speaker reduces the vowels in approximately a third of the cases (probably when he knows well the word pronunciation) and one beginner realizes one reduced vowel. For all other realizations the duration of the second vowel varies between 45 and 130 ms and its averaged duration with respect to other vowels of the word is approximately 0.8, for both groups. The averaged percentages of reduction were 7.5% and 2% for advanced learners and beginners, respectively.

4.1.2 Vowel duration ratio in `S1S2 S1`S2S3 words

Most speakers do not master L2 vowel durations (the averaged ratios are 1.05 and 0.85 for advanced speakers and beginners). Indeed, for all learners but one, the duration ratio (d(VS)/d(VUF)) is in between 0.7 and 1.1 (vs. 1.7 for English speakers). This confirms the tendency, well known for French learners, to lengthen the last syllable of the word and make relatively weak differences between syllable durations. The duration ratio is relatively high (1.5) and close to that of English speakers for only one speaker (an advanced one).

4.1.3 F0 pattern in `S1S2 and S1`S2S3 words

The realisation of this pattern seems to provide a good indication of the degree of proficiency of the subjects involved in this experiment. Let us first present the case of speakers with a low production level. We observe two kinds of pattern for these learners. Most speakers exhibit relatively flat patterns, with sometimes a tendency to rise F0 at the end of word, a behaviour typical of beginners and people apprehensive about speaking in a foreign language. One speaker exhibits systematically falling patterns. Since in the corpus, the accent never falls on the last syllable of the word, the percentage indicating the number of times the F0 maximum is on the right syllable is relatively low (33%). The difference in height, only taken into account when the F0 maximum is on the correct syllable, is weak (2.5 semi-tones). The advanced learners tend to exhibit correct F0 patterns, with the location of the F0 peak most of the times (90%) on the expected syllable and a substantial difference between the height of the stressed vowel and that of the post-stressed vowel (6.9 semi-tones).

4.1.4 Summary

To summarize the results of this first session, we remark strong differences in the mastering of F0 pattern and duration cues. Concerning duration, all speakers but one apply L1 duration rules when speaking English. Most of them are not aware of the reduction of post-stressed vowels, and others seem to encounter problems in predicting its occurence. On the contrary, some speakers, the more advanced ones, realize correct F0 patterns.

4.2 Second session. Effect of feedback for advanced learners

In this subsection, we examine the effet of feedback for advanced learners: auditory feedback (control condition) and diagnosis, F0 displays, as well as speech modification (test condition). There is no statistical difference between results for advanced speakers in test and control conditions so results for both groups are discussed at the same time.

4.2.1 Reduction phenomon in `S1S2S3 words

Feedback has a very high impact on advanced speakers, both in amplitude and degree of significance, whatever the condition. Indeed, in both condition, all speakers change their pronunciation to realize strongly reduced vowels for all words. We thus obtain a score of 100% for all speakers. It then appears here that auditory feedback was sufficient to enable advanced learners to improve their realization.

4.2.2 Vowel duration ratio in `S1S2 and S1`S2S3 words

All speakers who exhibit relatively low averaged ratios –i.e. all but one- improve themselves, whatever the condition, showing once more the high impact of auditory feedback for advanced learners. The averaged ratios are 1.2 and 1.25 in control and test conditions, respectively. This increase is mainly due to the lengthening of the stressed vowel, rather than a shortening of the last one.

4.2.3 F0 pattern in `S1S2 and S1`S2S3 words

We observed slight but significant increases (of about 0.6 semi-tones) in F0 relative height in both conditions. This might not be important on a perceptual point of view, but it seems that learners are aware of the important difference in height exhibited by the reference speaker between the stressed and the following unstressed vowel and try to imitate it. There was no more error in the location of F0 maximum.

4.2.4 Summary

Due to effect of feedback in both conditions, advanced speakers change the relative duration of the vowels, and reduce vowels that have been strongly reduced by reference speakers. For learners involved in this study, auditory feedback appears to be as efficient as more complex feedback such as the one proposed in this study. This result appears to be in agreement with that obtained by Herry and Hirst (2010) who tested advanced learners.

4.3 Second session. Effect of feedback for low production level speakers

4.3.1 Reduction phenomon in `S1S2S3 words

Auditory feedback does not have a high impact on low production level speakers. According to the comments they made after the experiment, they seem to have been disturbed by the way English speakers utter seemingly familiar words (transparent words) and slightly (but significantly) improved their realizations. On the whole, all speakers diminish the averaged duration of the second vowel but the averaged percentage of reduction is relatively low (37%).

Of course the effect of advanced feedback is far better. Indeed, the system detects the number of syllables and when this number is different from that of the reference, subjects were informed of this deviation and asked to change their pronunciation. Then, aware of what is expected, subjects make strong efforts to reduce vowels. The results varies with speakers and words: only one speaker reduces all vowels, but all speakers drastically reduce

V2 averaged duration (this reduction was highly significant). The average percentage of reduction was 63%. The difference between both conditions is highly significant.

4.3.2 Vowel duration ratio in `S1S2 and S1`S2S3 words

In control condition, there is no significant improvement with respect to vowel duration. The averaged duration ratio (d(VS)/D(UF)) estimated for all speakers stay in the same range as that observed in the first session.

In test condition, subjects significantly improved their realization, making generally longer stressed vowels (the averaged ratio rises from 0.84 up to 1.1). The difference between test and control condition is highly significant.

4.3.3 F0 pattern in `S1S2 and S1`S2S3 words

In control condition, the observed improvement is very small in amplitude (0.4 semi-tones) and significance and varies with speakers and words. Two speakers exhibit clear tendencies to raise the pitch of the stressed syllables with respect to the unstressed one, but this rise not systematic, i.e. not observed for all words. The overall percentage of words with the maximum of F0 on the right syllable increases from about 10%.

In test condition, there is a very significant improvement. All speakers were clearly informed of what was expected from them (thanks to small texts and arrows) and improved their realization most of the times (the percentage concerning the location of the F0 maximum raises from 30% up to 75%). The amplitude of the modification (1.4 semi-tones in the average) varies with speakers and words. Once more, the difference between test and control conditions is highly significant.

4.3.4 Summary

Due to effect of "advanced" feedback, subjects with a low oral production level significantly improve their realizations. Advanced feedback appears to more interesting than simple auditory feedback for these subjects.

5. Conclusion

Feedback on L2 prosody based upon visual displays, speech modifications and automatic diagnosis has been elaborated and a pilot experiment undertaken to test its immediate impact on listeners. Results show that the various kinds of feedback provided by the system enable French learners with a low production level to improve their realisations of English lexical accents more than (simple) auditory feedback. These results shoud be reinforced with a large number of speakers but based upon the important differences between results obtained for speakers in test and control conditions, we are confident in the interest of the system presented here. In particular, the system analyses learners' realisations and provide indications on what they should correct, a guidance which is considered as necessary by specialists in the oral aspects of language learning, such as Chun (1998) or Germain-Rutherford (Germain-Rutherford & Martin, 2000).

The perspectives of this work are twofold, technological and experimental. On a technological point of view, results from the experiment encourage us to pursue the work undertaken with the speech team ("Parole" team) at LORIA on speech alignment for non-native speakers. This work concerns both the detection of precise segment boundaries when possible and the elaboration of confidence levels on boundary detections. The design of such confidence levels would allow the system to propose feedback only when it can rely upon detections with high confidence levels and thus avoid erroneous corrections. We also plan to refine our method for modifying learners' voice. We use (an improved version) of TD-PSOLA method to modify segment durations and melodic curves. This algorithm does not allow us to modifiy vowel quality. Yet automatic modifications of vowel quality would be interesting to take into account vowel (timbre) reduction in post-stressed vowels. This modification could be obtained by using techniques such as voice morphing or techniques working in the frequency space. Since these techniques could generate degradations in speech quality, compromises should be found.

On an experimental point of view, it would be interesting to separate the impact of each kind of feedback (visual displays, diagnosis and perceptual feedback based upon speech modification). We also plan to investigate the long-term effect of automatic feedback should in collaboration with teachers in foreign language at the University of Lorraine.

6. Acknowledgment

This work has been partially funded through the INTERREG IVA ALLEGRO project (Project No.: 67 SMLW 1 1 137). We would like to thank Odile Mella for her help concerning the manuscript, the reference speakers, and the subjects participating in the experiment, as well as Yves Laprie, Guillaume Henry and Christian Gillot for their contribution to the software.

7. References

Bouselmi, G.; Fohr, D.; Illina, I. & Haton J.-P. (2005). Fully Automated Non-Native Speech Recognition Using Confusion-Based Acoustic Model Integration, *Proceedings of Interspeech* Lisbon, Portugal.

Chun, D. (1998). Signal analysis software for teaching discourse intonation. *Language learning and technology*. Vol. 2, No. 1, pp. 61-77.

Colotte, V.; Laprie, Y. & Bonneau, A. (2001). Percpetual experiments on enhanced and slowed down speech sentences for second language acquistion. *Proceedings on the international conference on speech communication and technology (ICSLP),* Aalborg, Denmark.

Colotte, V. & Laprie, Y. (2002). Higher pitch marking precision for TD-PSOLA, *Proceedings of European Signal Processing Conference (EUSIPCO),* Toulouse.

Dauer, R. (1983). Stress-timing and syllable-timing reanalysed. *Journal of Phonetics,* Vol. 11, pp. 51–62.

De Bot, K. (1983). Visual feedback on intonation I: Effectiveness and induced practice behaviour. *Language and Speech* , Vol.6 , No.4, pp. 331–350

Hazan, V. & Simpson, A. (1998). The effect of cue-enhancement on the intelligibility of nonsense word and sentence materials presented in noise," *Speech Communication*, vol. 24, pp. 211–226, 1998.

Fohr, D.; Mari, J. F. & Haton, J. P. (1996). Utilisation de modèles de Markov pour l'étiquetage automatique et la reconnaissance de BREF80, *Journées d'Etude de la Parole,* Avignon, France.

Germain-Rutherford, A. & Martin, P. (2000). Utilisation d'un logiciel de visualisation pour l'apprentissage de l'oral en langue seconde, *Apprentissage des Langues et Systèmes d'Information et de Communication (ALSIC),* Vol. 3, No.1, pp. 71-86.

Henry, G. ; Bonneau, A. & Colotte, V. (2007). Tools devoted to the acquisition of the prosody of a foreign language, *Proceedings of the International Congress of Phonetic Sciences*, Saarbrücken, Germany.

Herry, N. & Hirst, D. (2010). Subjective and Objective Evaluation of the Prosody of English Spoken by French Speakers: the Contribution of Computer Assisted Learning, *Speech Prosody*, Aix en Provence, France.

Ingram, J.; Mixdorff, H. & Kwon, N. (2009). Voice morphing and the manipulation of intra-speaker and cross-speaker phonetic variation to create foreign accent continua: A perceptual study, *SLaTE 2009 ISCA International Workshop on Speech and Language Technology in Education*, Wroxall Abbey Estate, Warwickshire, England. 3-5 September 2009.

James, E. (1977). The Acquisition of a Second-Language intonation Using a Visualizer. *Canadian Modern Language Review*, Vol.33, No.4, pp. 503-506.

Kommissarchik, J. & Kommissarchik, E. (2004). BetterAccent Tutor- Analysis and Visualization of Speech Prosody. *Proceedings of InSTIL/ICALL,* Dundee, Scotland.

Laprie, Y. (1999). Snoori, a software for speech sciences. *Proceedings of MATISSE*, London, England.

Loizou, P. (1998). Mimicking the human ear. *IEEE Signal processing magazine*. September 1998 pp. 101-130.

Martin, P. (2004). WinPitch LTL II, a Multimodal Pronunciation Software. *Proceedings of InSTIL/ICALL,* Venice, Italy.

Moulines, E. & Charpentier, F. (1990). Pitch synchronous wave-form processing techniques for a text-to-speech synthesis using diphones. *Speech Communication,* Vol.9, No.5-6, pp. 453-467.

Ortega, M. and Hazan, V. (1999). Enhancing acoustic cues to aid L2 speech perception. *Proceedings of the International Congress of Phonetics Sciences,* San Fransico, California. pp. 117-120.

Tallal, P.; Miller, S.; Bedi, G.; Byma, G.; Wang, X.; Nagarajan, S.; Schreiner, C.; Jenkins, W. & Merzenich, M. (1996). Language comprehension in language-learning impaired children improved with acoustically modified speech, *Science*, vol. 271, pp. 81–84.

Vardanian, R. (1964).Teaching English through oscilloscope displays, *Language Learning*, Vol.14, No. 3-4, pp. 109-117.

Yanagisawa, K. & Huckvale, M. (2007). Accent morphing as a technique to improve the intelligibility of foreign-accented speech, *Proceedings of the International Congress of Phonetics Sciences,* Saarbrücken, Germany.

www.tellmemore.com
Auralog
www.langmaster.com
LANGMaster
www.loria.fr/~laprie/WinSnoori
Yves Laprie
http://www.interreg-4agr.eu/
INTERREG IVA ALLEGRO project

5

Exploring Speech Technologies for Language Learning

Rodolfo Delmonte
Università Ca' Foscari - Ca' Bembo,
Linguistic Computational Laboratory,
Italy

1. Introduction

The teaching of the pronunciation of any foreign language must encompass both segmental and suprasegmental aspects of speech. In computational terms, the two levels of language learning activities can be decomposed at least into phonemic aspects, which include the correct pronunciation of single phonemes and the co-articulation of phonemes into higher phonological units; as well as prosodic aspects which include

- the correct position of stress at word level;
- the alternation of stress and unstressed syllables in terms of compensation and vowel reduction;
- the correct position of sentence accent;
- the generation of the adequate rhymth from the interleaving of stress, accent, and phonological rules;
- the generation of adequate intonational pattern for each utterance related to communicative functions;

As appears from above, for a student to communicate intelligibly and as close as possible to native-speaker's pronunciation, prosody is very important [2.]. We also assume that an incorrect prosody may hamper communication from taking place and this may be regarded a strong motivation for having the teaching of Prosody as an integral part of any language course.From our point of view it is much more important to stress the achievement of successful communication as the main objective of a second language learner rather than the overcoming of what has been termed "foreign accent", which can be deemed as a secondary goal. In any case, the two goals are certainly not coincident even though they may be overlapping in some cases. We will discuss about these matter in the following sections.

All prosodic questions related to "rhythm" will be discussed in the first section of this chapter. In [62.] the author argues in favour of prosodic aids, in particular because a strong placement of word stress may impair understanding from the listener's point of view of the word being pronounced. He also argues in favour of acquiring correct timing of phonological units to overcome the impression of "foreign accent" which may ensue from an incorrect distribution of stressed vs. unstressed stretches of linguistic units such as syllables or metric feet. Timing is not to be confused with speaking rate which need not be increased forcefully to give the impression of a good fluency: trying to increase speaking rate may result in lower intelligibility. The question of “foreign accent” is also discussed at length in (Jilka M., 1999). This work is particularly relevant as far as intonational features of a learner of a second language which we will address in the second section of this chapter. Correcting the Intonational Foreign Accent (hence IFA) is an important component of a Prosodic Module for self-learning activities, as categorical aspects of the intonation of the two languages in contact, L1 and L2 are far apart and thus neatly distinguishable. Choice of the two languages in contact is determined mainly by the fact that the distance in prosodic terms between English and Italian is maximal, according to (Ramus, F. and J. Mehler, 1999; Ramus F., et al., 1999).

1.1 Speech recognition and acoustic models

In all systems based on HMMs (Kawai G., K.Hirose, 1997; Ronen O. et al., 1997), student's speech is segmented and then matched against native acoustic models. The comparison is done using HMM loglikelihoods, phone durations, HMM phone posterior probabilities, and a set of scores is thus obtained. They should represent the degree of match between non-native speech and native models. In the papers quoted above, there are typically two databases, one for native and another for nonnative speech which are needed to model the behaviour of HMMs. As regards HMMs, in (Kim Y., et al. 1997) the authors discuss the procedure followed to generate them: they are trained on the native speakers database where dynamic time warping has applied in order to eliminate the dependency of scoring for each phone model on actual segment duration. Duration is then recovered for each phone from each frame measurements and normalized in order to compensate for rate of speech. Phonetic time alignment is then automatically generated for the student's speech.
HMModels are inappropriate to cope with prosodic learning activities since theymay produce distorted results in a teaching environment. This may be due, first of all, to the fact that they produce a set of context-independent models for all phone classes and this fact goes against the linguistically sound principle that says that learning a new phonological system can only be done in a context-dependent fashion. Each new sound must be learnt in its context, at word level, and words should be pronounced with the adequate prosody, where duration plays an important role. One way to cope with this problem would be that of keeping the amount of prosody to be produced under control: in other words to organize tasks which are prosodically "poor" in order to safeguard students from the teaching of bad or wrong linguistic habits.Then there is the well-know problem of the quantity of training data to be used to account for both inter-speaker and intra-speaker variability. In addition, since a double database should be used, one for native and one for non-native speakers, the question is what variety of native and non-native is being chosen, seen that standard pronunciation is an abstract notion. As far as prosody is concerned, we also know that there is a lot of variability both at intraspeaker and interspeaker level: this does not hinder efficient and smooth communication from taking place, but it may cause problems in case of a student learning a new language. Other problems are related to well-known unsuitability of HMM to encode duration seen that this parameter cannot be treated as an independent variable (but see the discussion in the sections below). Other non-independent aspects regard transitions onto and from a given phonetic segment together with carryover effects due to the presence of previous syllabic nasal or similar sonorant units. In addition, the maximum likelihood estimate and smoothing methods introduce errors in each HMM which may be overlooked in the implementation of ASR systems for dictation purposes; but

not in the assessment of Goodness of Pronunciation for a given student with a given phoneme. Generally speaking, HMMs will only produce decontextualized standard models to follow for the student, which are intrinsically unsuited to be used for assessment purposes in a teaching application.

In pronunciation scoring, technology is used to determine how well the expected word/utterance was said. It is simple to return a score; the trick is to return a score that "means" something (Price P., 1998:105). Many ASR systems have a score as a by-product. However, this score is tuned for use by native speakers, and does not tend to work well for language learners. Therefore, unacceptable or unintelligible utterances may receive good scores (false positives), and intelligible utterances may receive poor scores (false negatives).

SLIM makes use of Speech Recognition in a number of tasks which exploit it adequately from the linguistic point of view. We do not agree with the use of speech recognition as adequate assessment tool for the overall linguistic competence of a student. In particular, we do not find it suited for use in language practice with open-ended dialogues given the lack of confidence in the ability to discriminate and recognize Out-Of-System utterances (Meador J., 1998). We use ASR only in a very controlled linguistic context in which the student has one of the following tasks:

- repeat a given word or utterance presented on the screen and which the student may listen to previously - the result may either be a state of recognition or a state of non-recognition. The Supervisor will take care of each situation and then allow the student to repeat the word/utterance a number of times;
- repeat in a sequence "minimal pairs" presented on the screen and which the student may listen to previously - the student has a fixed time interval to fulfil the task, and a certain number of total possible repetitions (typically twenty) - at the end, feedback will be number of correct repetitions;
- speak aloud one utterance from a choice among one to three utterances appearing on the screen as a reply to a question posed by a native speaker's voice or by a character in a video-clip. This exercise is called Questions and Answers and is usually referred to a False Beginner-Intermediate level of proficiency of the language. The student must be able to understand the question and to choose the appropriate answer on the basis of grammatical/semantic/pragmatic information available. The outcome may be either a right or a wrong answer, and ASR will in both cases issue the appropriate feedback to the student;
- do role-play, i.e. intervene in a dialogue of a video-clip by producing the correct utterance when a red light blinks on the screen, in accordance with a given communicative function the student is currently practising. This is a more complex task which is only allowed to be accessed by advanced students: the system has a number of alternative utterances connected with each communicative function the student has to learn. The interaction with the system may be both in real time or in slow-down motion: in the second case the student will have a longer time to synchronize his/her spoken utterance with the video-clip.

One might question the artificiality of the learning context by reminding the well-known fact that a language can only be learnt in a communicative situation (Price P., 1998). However we feel that the primary goal of speech technology is to help the student develop good linguistic habits in L2, rather than engaging the student in the use of "knowledge of the world/context" creatively in a second language. Thus we assume that speech technology should focus on teaching systems which incorporate tools for prosodic analysis focussing on the most significant acoustic correlates of speech in order to help the student imitate as close as possible the master performance, contextualized in some communicative situation.

Some researcher have tried to cope with the problem of identifying errors in phones and prosody within the same ASR technology (Eskénazi, M., 1999). The speech recognizer in a "forced alignment mode" can calculate the scores for the words and the phones in the utterance. In forced alignment, the system matches the text of the incoming signal to the signal, using information about the signal/linguistic content that has already been stored in memory. Then after comparing the speaker's recognition scores to the mean scores for native speakers for the same sentence pronounced in the same speaking style, errors can be identified and located (Bernstein, J., & Franco, H., 1995). On the other hand, for prosody errors, duration can be obtained from the output of most recognizers. In rare cases, fundamental frequency may be obtained as well. In other words, when the recognizer returns the scores for phones, it can also return scores for their duration. On the other hand, intensity of the speech signal is measured before it is sent to the recognizer, just after it has been preprocessed. It is important that measures be expressed in relative terms - such as duration of one syllable compared to the next - since intensity, speaking rate, and pitch vary greatly from one individual to another.

The FLUENCY system – which will be illustrated further on in the chapter - uses the SPHINX II recognizer to detect the student's deviations in duration compared to that of native speakers. The system begins by prompting the student to repeat a sentence. The speech signal and the expected text are then fed to the recognizer in forced alignment mode. The recognizer outputs the durations of the vowels in the utterance and compares them to the durations for native speakers. If they are found to be far from the native values, the system notifies the user that the segment was either too long or too short.

In Bagshaw et al. (1993) student's contours are compared to those of native speakers in order to assess the quality of pitch detection. Rooney et al. (1992)applied this to the SPELL foreign language teaching system and attached the output to visual displays and auditory feedback. One of the basic ideas in their work was that the suprasegmental aspects of speech can be taught only if they are linked to syllabic information. Pitch information includes pitch increases and decreases and pitch anchor points (i.e., centers of stressed vowels). Rhythm information shows segmental duration and acoustic features of vowel quality, predicting strong vs. weak vowels. They also provided alternate pronunciations, including predictable cross-linguistic errors.As we will argue extensively in this part of the Chapter, we assume that segmental information is in itself insufficient to characterize non-native speech prosody and to evaluate it. In this respect, "forced alignment mode" for an ASR working at a segmental/word level still lacks hierarchical syllabic information as well as general information on allowable deviations from mother-tongue intonation models which alone can allow the system to detect prosodic errors with the degree of granularity required by the application.

2. Section I: Prosodic tools for self-learning activities in the domain of rhythm

2.1 General problems related to Rhythm

In prosodic terms, Italian/Spanish and English are placed at the two opposite ends of a continuum where languages of the world are placed (Ramus, F. and J. Mehler, 1999; Ramus

F., et al. 1999). This is dependent on their overall phonological systems, which in turn are bound by the vocabulary of the languages. The Phonological system will typically determine the sound inventory available to speakers of a given language; the vocabulary will decide the words to be spoken. The Phonological system and the vocabulary in conjunction will then determine the phonotactics and all suprasegmental structures and features.

As far as syllables are concerned, we should also note that their most important structural component, the nucleus, is a variable entity in the two language families: syllable nuclei can be composed of just vowels or of vowels and sonorants. Vowel and sonorant sounds being similar would account for the greatest impression of two languages sounding the same or very close: from a simplistic segmental point of view, English and Italian/Spanish would seem to possess similar prosodic behaviour as far as sonorants are concerned. On the contrary, we should note the fact that English would syllabify a sonorant as syllable nucleus – as would German – but this would be totally unknown to a Romance Italian/Spanish speaker. Contrastive studies have clearly pointed out the relevance of phonetic and prosodic exercises both for comprehension and perception. In general prosodic terms, whereas the prosodic structure of Italian is usually regarded as belonging to the syllable-timed type of languages, that of English is assumed to belong to stress-timed type of languages (Bertinetto P.M., 1980; Lehiste I., 1977). This implies a remarkable gap especially at the prosodic level between the two language types. Hence the need to create computer aided pronunciation tools that can provide appropriate feedback to the student and stimulate pronunciation practice.

Reduced vowels typically affect duration of the whole syllable, so duration measurements are usually sufficient to detect this fact in the acoustic segmentation. In stress-timed languages the duration of interstress intervals tends to become isochronous, thus causing unstressed portions of speech to undergo a number of phonological modifications detectable at syllable level like phone assimilation, deletion, palatalization, flapping, glottal stops, and in particular vowel reduction. These phenomena do not occur in syllable-timed languages - but see below - which tend to preserve the original phonetic features of interstress intervals (Bertinetto P.M., 1980). However a number of researcher have pointed out that isochrony is much more a matter of perception than of production (see in particular, Lehiste I., 1977). Differences between the two prosodic models of production are discussed at length in a following section.

2.1.1 Segmental vs. syllable-based modeling

Prosodic data suffer from a well-known problem of sparsity (Delmonte R., 1999). In order to reach a better understanding of this problem however, we would like to comment on data in the literature (van Son R., J. van Santen, 1997; Umeda N., 1977; van Santen J., 1997) basically related to English, apart from the latter, and compare them with data available on Italian. We support the position also endorsed by Klatt and theoretically supported by Campbell and Isard in a number of papers (Campbell W., S.Isard, 1991; Campbell W., 1993), who consider the syllable the most appropriate linguistic unit to refer to in order to model segmental level phonetic and prosodic variability.

The reason why the coverage of data collected for training corpus is disappointing is not simply a problem of quantities, which can be solved by more training data. The basic problem seems to be due to two ineludible prosodic factors:

- the need to encode structural information in the syllable, which otherwise would belong to higher prosodic units such as the Metric Foot, The Clitic Group, The Phonological Group (which will be discussed in more detail below);
- the prosodic peculiarity of the English language at syllable level.

I am here referring to the great variety of syllabic nuclei available in English due to the high number of vowels and diphthongs and also to the use of syllabic consonants like nasals or liquids as syllable nuclei. the presence of a too large feature space, or too great number of variables to be considered. When compared with a language like Chinese, we see two languages at the opposite sides: on the one side a language like Chinese where syllables have a very limited distribution within the word and a corresponding limitation in the type of co-occuring vowel; on the other side very high freedom in the distribution of syllables within the word as our data will show. As to stressed vs unstressed syllables the variability is very limited in Chinese due to the number of stressable vowels, and also due to the fact that most words in Chinese are monosyllabic. In addition, syllable structure is highly simplified by the fact that no consonant clusters are allowed. In fact (van Santen J., et al., 1997:321; Grover et al., 1998) reports the number of factors and parameters used to compute the multilingual prosodic model for Chinese, French and German we see that Chinese has less than one third the number of classes and less than half the number of parameters than the other two languages. English, which is not listed, is presented in (van Son R., J. van Santen, 1997) with the highest number of factors, 40. Sparsity in prosodic data is then ultemately linked to the prosodic structure of the language, which in turn is partly a result of the interaction between the phonological and the lexical system of the language.

2.1.2 Evaluation tools for timing and rhythm

As stated in the Introduction, assessment and evaluation are the main goal to be achieved by the use of speech technology, in order to give appropriate and consistent feedback to the student. Theoretically speaking, assessment requires the system to be able to decide at which point in a graded scale the student's proficiency is situated. Since students usually develop some kind of interlingua between two opposite poles, non-native beginners and full native pronunciation, the use of two acoustic language models should be targeted to low levels of proficiency, where performance is heavily encumbered, conditioned by the attempts of the student to exploit L1 phonological system in learning L2. This strategy of minimal effort will bring as a result a number of typical errors witnessing to a partial overlapping between the two concurrent phonetic inventories: phonetic substitutes, for phonetic classes not attested in L2 will cause the student to produce words which only approximate the target sound sequence perhaps by manner but not by place of articulation as is the usual case with dental fricatives in English [ð, θ]. Present-day speech recognizers are sensitive exclusively to phonetic information concerning the words spoken - their contents in terms of single phones. Phonetically based systems are language-specific, not only because the set of phonemes is peculiar to the language but also because the specification of phonetic context means that only certain sequences of phonemes can be modeled. This presents a problem when trying to model defective pronunciations generated by non-native speakers. For example, it might be impossible to model the pronunciation [zæt] - typical of languages lacking dental fricatives - for the word *that* with a set of triphones designed only for normal English pronunciations.

Current large-vocabulary recognition systems use *sub-word* reference model units at the phoneme level. The acoustic form of many phonemes depends critically on their phonetic context, particularly the immediately preceding and following phonemes. Consequently, almost all practical sub-word systems use *triphone* units; that is, a phoneme whose neighbouring phoneme to the left and to the right is specified. Clearly, only in case some errors are detected and evaluated, the system may try to guess which level of interlingua the student belongs to.Thus the hardest task ASR systems are faced with is segmentation. In Hiller et al. (1993) segmentation is obtained using a HMM technique where the labeling of the incoming speech is constrained by a segmental transition network which is similar to our lexical phonetic description in terms of phones with associated phonetic and phonological information. In their model however, a variety of alternative pronunciations are encoded, including errors predictable from the student's mother tongue. These prediction are obtained from a variety of different sources (see ibid., 466). In our case, assessment of the student's performance is made by a comparative evaluation of the expected contrastive differences in the two prosodic models in contact, L1 and L2.

As Klaus Zechner, et al. (2009) comment, while speech scoring systems for linguistically simpler tasks such as reading or providing a short response have been in operation for some time (Bernstein J., 1998, 1999; H. Franco et al.), few attempts have been made to automatically score spontaneous, non-native speech where the term 'spontaneous' is referred to high entropy speech where a large-vocabulary continuous speech recognition (LVCSR) system needs to be used for recognizing speakers' utterances. ETS has, after several years of research (see K. Zechner, I. I. Bejar, and R. Hemat), designed and implemented an operational system, SpeechRater™, for scoring spontaneous non-native speech in the context of the TOEFL® iBT Practice Online (TPO) Speaking practice program. In the currently operational Version 1, however, the main area of feature coverage is fluency. The architecture of the SpeechRater system is a concatenation of these three components: a arge-vocabulary continuous speech recognition (LVCSR) system trained on non-native speech, a feature computation module, and a multiple regression scoring module.The interesting point is that the speech recognizer has been trained on "non-native" speech: in particular 30 hours of speech have been used and 100 hours for the language model training. The ASR then computes a total of 40 features which are appropriate for the task and their usage fits well with human raters' judgements.

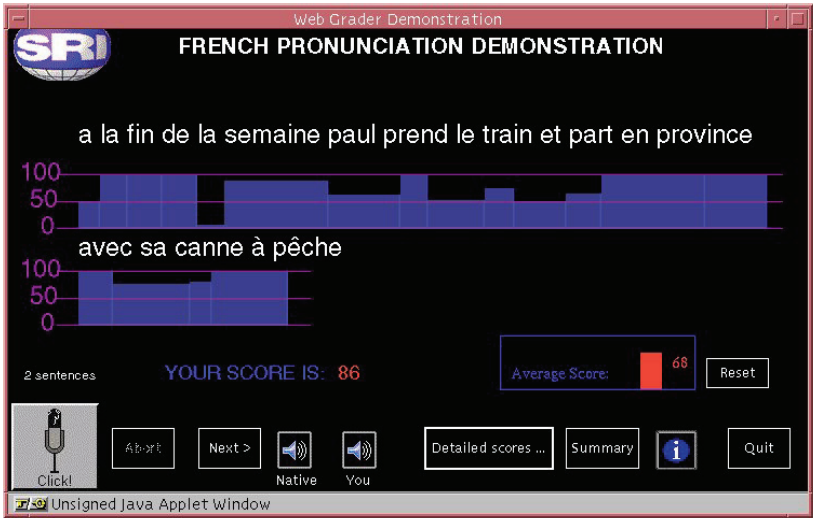
C. Cucchiarini, S. Strik, and L. Boves (1997a) and C. Cucchiarini, S. Strik, and L. Boves (1997b) describe a system for Dutch pronunciation scoring along similar lines. Their feature set, however, is more extensive and contains, in addition to log likelihood Hidden Markov Model scores, various duration scores, and information on pauses, word stress, syllable structure, and intonation. In an evaluation, correlations between four human scores and five machine scores range from 0.67 to 0.92.In a more recent paper on an algorithm called the Goodness of Pronunciation, Sandra Kanters, Catia Cucchiarini, Helmer Strik compile an inventory of pronunciation errors frequently made by foreigners speaking Dutch. On the basis of this inventory they create artificial errors in a native development corpus, which in turn were used to optimize thresholds for the Goodness of Pronunciation (GOP) algorithm, which they use to give corrective feedback to users at the phoneme level.As the authors comment, in pronunciation learning corrective feedback is particularly required because very often learners are not awareof the pronunciation errors they make. Since exposure to the L2 and L2 output will not automatically guarantee this kind of awareness, corrective feedback is required to make learners aware of their errors and stimulate them to attempt self-improvement (Havranek, G.).

2.2 State of the art in CALL tools: rhythm

Here below we list and briefly present those CALL systems that are located on the web which have tackled the problem of student's assessment in the field of word and subword syllable units using automatic visualization and correction methods. The comments and pictures are taken from the website of come from a publication of the author.

2.2.1 WebGrader

WebGraderTM (Neumeyer L., et al, 1998) is a pronunciation grading tool designed for practicing pronunciation in a second language. The system uses SRI's speech recognition and pronunciation scoring technologies. The application client was implemented by using the Java platform to facilitate deployment and updates of software and content over the World Wide Web. We present the overall system architecture, user-interface design, scoring algorithms, and a preliminary user study. WebGraderTM is organized in lessons. A lesson is a collection of related sentences organized by themes such as transportation or eating in a restaurant. Students can listen to natives saying the phrases, part of the phrases, or individual words. They can also record themselves and obtain pronunciation scores for the phrase and for individual words. Words that are hard to produce can be practiced by selecting the target word and obtaining scores for that particular word. The content can easily be updated, and additional lessons can be downloaded from a content server.



Web Grader Demonstration

FRENCH PRONUNCIATION DEMONSTRATION

a la fin de la semaine paul prend le train et part en province

avec sa canne à pêche

2 sentences

YOUR SCORE IS: 86

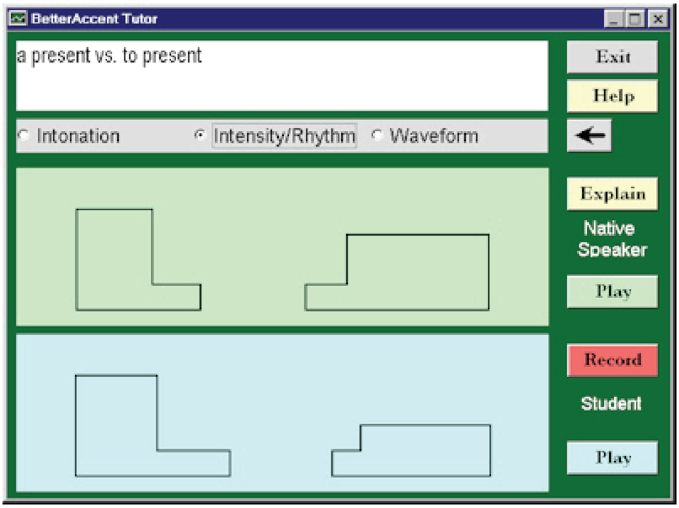
Average Score: 68

Fig. 1. WebGrader Visualization of graded pronunciation of French utterance

2.2.2 BetterAccent Tutor for English

BetterAccent Tutor (Komissarchik E., Julia Komissarchik, 2000a, 2000b) is designed for non-native speakers of English, who would like to speak clearly, effectively and be easily understood. Using advance unique patented speech analysis technology, BetterAccent Tutor presents instant audio-visual feedback of users' pronunciation. In American English three components of speech that contribute the most to comprehensibility are *intonation, stress and rhythm*. BetterAccent Tutor analyzes intonation, stress and rhythm patterns of a user-recorded utterance and visualizes these patterns in an easy- to- understand manner. By pinpointing the exact mistakes, BetterAccent Tutor allows users to focus on the problems that are unique to their speech. It allows users to *record*and*playback* utterances. Analyzes and visualizes*intonation, intensity* and *rhythm* patterns of recorded utterances. Visualizes *the syllabic structure* of recorded utterances and *highlights the syllables* as they are played back. Allows users to *visually compare* the user's and native speaker's *intonation*, *intensity* and *rhythm* patterns. Contains an *extensive set of exercises* specially- designed for the BetterAccent Tutor. Includes *detailed explanations* of each exercise. Includes a *large collection of utterances by native speakers* to provide users with guidance and a yardstick for correct pronunciation. Works well as a course supplement or as an interactive pronunciation coach for students' independent study.

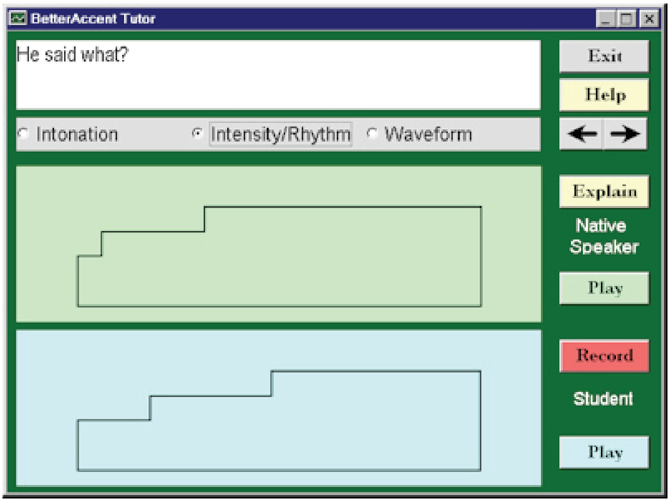
BetterAccent Tutor's purpose is to help students speak clearly and effectively and to be easily understood. We believe that there is no such thing as right or wrong pronunciation; not even two native speakers speak alike. But to be understood by native and non-native speakers, it is imperative for non-native speakers to match native speakers at certain key points. With visual feedback, the Tutor shows users' speech characteristics that are most important. As commented above, the three factors that have the biggest impact on intelligibility of speech are intonation, stress and rhythm. BetterAccent Tutor analyzes and visualizes intonation, stress and rhythm patterns of users' speech. By visualizing users' pronunciation, the Tutor allows users to focus on the problems that are unique to their speech. The Tutor is designed to give users the power to identify, understand and correct pronunciation errors. BetterAccent Tutor Comprehensive Curriculum includes: Word Stress; Simple Statements; Wh-Questions; General Questions; Repeated Questions; Alternative Questions; Tag Questions; Commands; Exclamations; Direct Address; Series of Items; Long Phrases; Tongue Twisters



a present vs. to present

○ Intonation ● Intensity/Rhythm ○ Waveform

Fig. 2. BetterAccent Visualization of word stress example



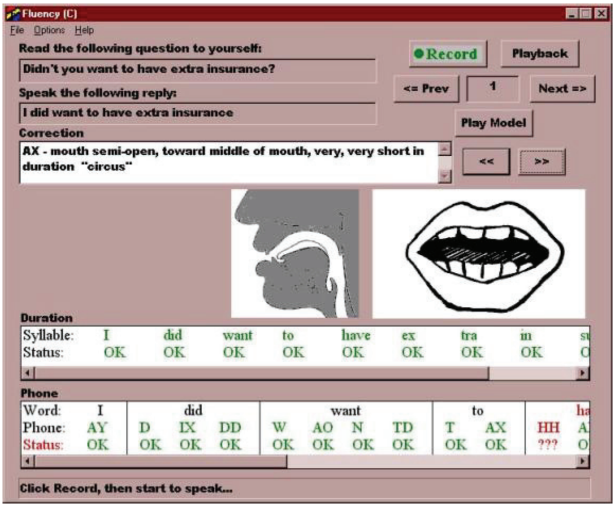
He said what?

○ Intonation ● Intensity/Rhythm ○ Waveform

Fig. 3. BetterAccent Visualization of utterance example

2.2.3 Fluency

The FLUENCY (Eskénazi, M., 1999; Eskénazi M., et al. 2000) project has investigated the detection of changes in duration, amplitude, and pitch that can reliably detect where non-native speakers deviate from acceptable native values, independently of L1 and L2. Thus, if a learning system is applied to a new target language, its prosody detection algorithms do not have to be changed in any fundamental way. Since they are separate from one another, the three aspects of prosody can easily be sent to visual display mechanisms that show how to correctly produce pitch, duration, or amplitude changes as well as compare a native speaker's production to that of a non-native speaker.



Read the following question to yourself:

Didn't you want to have extra insurance?

Speak the following reply:

I did want to have extra insurance

Correction

AX - mouth semi-open, toward middle of mouth, very, very short in duration "circus"

Duration

Syllable:	I	did	want	to	have	ex	tra	in	su
Status:	OK	OK	OK	OK	OK	OK	OK	OK	O

Phone

Word:	I		did			want			to		ha
Phone:	AY	D	IX	DD	W	AO	N	TD	T	AX	HH	A
Status:	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	???	O

Click Record, then start to speak...

Fig. 4. FLUENCY Visualization of utterance example

2.2.4 Ordinate's PhonePass

Ordinate's patented PhonePass® (Bernstein J., 1998; Bernstein J., et al. 1998) testing system is based on of years of research in speech recognition, statistical modeling, linguistics, and testing theory. The technology uses a speech recognition system that is specifically designed to analyze speech components from native and non-native speakers of English. In addition to recognizing words, the system also locates and evaluates relevant segments, syllables, and phrases in speech. The PhonePass system then uses statistical modeling techniques to assess the spoken performance.Independent studies have shown that Ordinate's SET tests (Spoken English Tests), which are powered by the PhonePass testing system, are more objective and reliable in operation than today's best human-rated tests, including one-on-one oral proficiency interviews. Using criteria developed by expert linguists, the PhonePass testing system provides items and scores that have been validated with reference to human judgments of proficiency, fluency, and pronunciation.

The PhonePass testing system uses speech recognition technology that was built to handle the different rhythms and varied pronunciations used by native and non-native English speakers. The system generates scores based on the exact words used in the spoken responses, as well as the pace, fluency, and pronunciation of those words in phrases and sentences. In addition to recognizing the words uttered, the system also aligns the speech signal, i.e., it locates the part of signal containing relevant segments, syllables, and words.Base measures are then derived from the linguistic units (segments, syllables, words), based on statistical models of native speakers. The base measures are combined into four diagnostic sub-scores using advanced statistical modeling techniques. Two of the diagnostic sub-scores are based on the content of what is spoken, and two are based on the manner in which the responses are spoken. An Overall Score is calculated as a weighted combination of the diagnostic sub-scores.

For the SET-10 test, responses to four item tasks are currently used for automated scoring. These are: reading aloud, repeating sentences, building sentences, and giving short answers to questions. In scoring, there is exactly one correct word sequence expected for each response to the reading and repeat items. Expert judgment was used to define correct answers to the short-answer question and sentence-build items. Most of the short-answer and some of the sentence-build items have multiple answers that are accepted as correct. All short-answer questions were pre-tested on diverse samples of native and non-native speakers. All items retained in the item banks were answered correctly by at least 90% of the native sample.

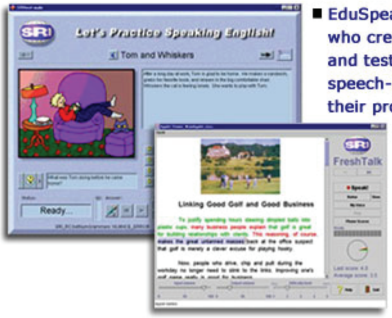
2.2.5 SRI's EduSpeak

EduSpeak ® (Franco H., et al., 2000) is a speech recognition system that, through its Software Development Kit, enables developers of multimedia applications to incorporate continuous speaker-independent speech recognition into their applications.Developed in the Speech Technology and Research (STAR) Laboratory of the Information and Computing Sciences Division at SRI International, EduSpeak® is now available for licensing in the Language Education, Reading Development, and Corporate Training markets. Interactive English as Second Language (ESL) instructional CDs for elementary school children, using EduSpeak®'s unique pronunciation scoring technology

- Computer-aided collection and grading of spoken language in education and corporate settings
- Multimedia edutainment software with speech enhanced interactivity
- Language training courses for corporate travelers

Features & Benefits:

Speaker independence: No user training required. Continuous speech capability: No need for artificial pauses. State-of-the-art performance: High level of accuracy. Compact engine and models downloads: Fast application loading and internet. Multiple native speech models: Multiple language capability. Non-native speech models: Robustness to strong accents. Children's speech models: Increased accuracy for children. Pronunciation grading capability: Pronunciation feedback. Dynamically loadable vocabulary: Application flexibility. Arbitrary grammars: Increased flexibilty in task design. Dynamically loadable grammars task: Dynamic configuration of recognizer

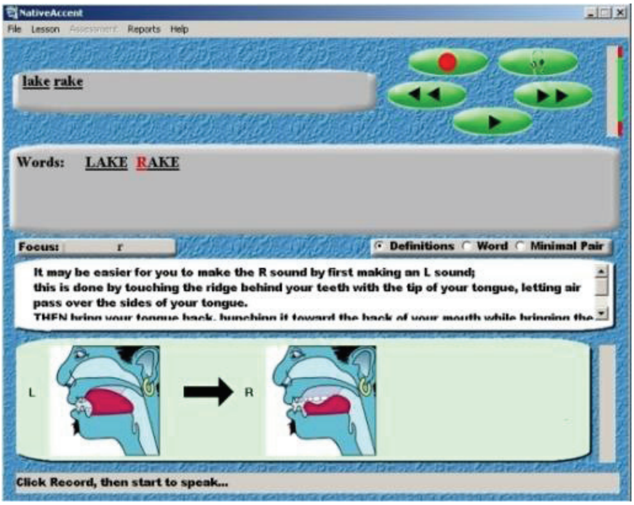

SRI
Let's Practice Speaking English!
Tom and Whiskers
Ready
Linking Good Golf and Good Business
FreshTalk

- EduSpeak® enables businesses who create instructional, training, and testing software to incorporate speech-based interactivity into their products
- The EduSpeak® software development toolkit gives developers a compact, yet state-of-the-art speech recognition engine, a choice of authoring interfaces, and support for multiple operating systems

Fig. 5. EduSpeak website advertisement

2.2.6 CMU Native Accent

NativeAccentTM(Eskénazi, M., 2007) is a pronunciation tutor using automatic speech recognition from the CMU. It has gone through a full-fledged assessment by real users in real situations, based on the customer's own criteria instead of more academic measures, and the variations in the customers' measures. Results in one study show that subjects who used NativeAccentTM did more than twice as well as the control group while both groups had human instruction.



lake rake

Words: LAKE RAKE

Focus: r

● Definitions ○ Word ○ Minimal Pair

It may be easier for you to make the R sound by first making an L sound;
this is done by touching the ridge behind your teeth with the tip of your tongue, letting air pass over the sides of your tongue.
THEN bring your tongue back, bunching it toward the back of your mouth while bringing the

Click Record, then start to speak...

Fig. 6. Main screen of NativeAccent showing feedback

The system has been implemented for pronunciation error detection but also for a complete course of study, with leveled corrective feedback information, a curriculum, a student model, astrategy on how to proceed through the curriculum for differentlearners (fast and slow, for example) and a reporting mechanism forthe teacher (to follow individual and grouped student progress).

2.3 Self-aearning activities in the prosodic module: word stress and timing

We now present*SLIM* an interactive multimedia system for self-learning of foreign languages which is currently addressed to Italian speakers. It has been developed partly under HyperCard™, and partly under MacroMedia Director™. However at present, the Prosodic Module interacts in real time only with HyperCard™ [24].

2.3.1 Preprocessing phase and timing modeling

As far as prosodic elements are concerned, prosodic evaluation is at first approximated from a dynamic comparison with the Master version of the current linguistic item to practice. In order to cope with L1 and L2 on a fine-grained scale of performance judgement, we devised and used in our system two types of models:

MODEL I: - Top-down Syllable-based Model for Syllable-Timed languages

It is a model in which durational structure for a phonological or an intonational phrase is specified first, and then the segmental duration of the grammatical units in the words are chosen as to preserve this basic pattern. The pattern is very well suited for syllable-timed languages, in which the number of syllables and the speaking rate could alone determine the overall duration to be distributed among the various phonetic segments according to phonological and linguistic rules. Mean values for unstressed and stressed syllables could be assigned and then refurbished according to number of phones, their position at clause and phrase level, their linguistic and informational role. Lengthening and shortening apply to mean durational values of segmental durations. In a partial version of this Model, inherent consonant durations are applied at general phonetic classes in terms of compressibility below/above a certain threshold and not at single segments. Since variability is very high at segment level, we apply an "elasticity" model (Campbell W., S.Isard, 1991; Campbell W., 1993) which uses both position and prosodic type to define minima and maxima, and then compute variations by means and standard deviations.

MODEL II: - Bottom-up Segment-based Model for Stress-Timed languages

In this model the starting point is the assignment of inherent duration to each phonetic segment which is followed by use of phonological rules to account for segmental interactions and influences of higher-level linguistic units. For English, Klatt (1987:132) chooses this model which reflects a bias toward attempting to account for durational changes due to local segmental environment first, and then looking for any remaining higher level influences. In this model, the relative terms lengthening and shortening of the duration of a segment has sense if related to inherent duration for a particular segment type. The concept of a limiting minimum duration or equivalently the incompressibility can be better expressed by beginning with the maximum segmental duration (Klatt, D., 1987:132). In fact, we resort again to the "elasticity" hypothesis at syllable level, since we found that working at segmental level does not produce adequate predictions.

2.3.2 Segmentation and stress marking

Consider now the problem of the correct position of stress at word level and the corresponding phenomena that affect the remaining unstressed syllables of words in English. First of all, prominence at word level is achieved by increased duration and intensity and/or is accompanied by variations in pitch and vowel quality (like for instance vowel reduction or even deletion, in presence of syllabifiable consonant like "n, d"). To detect this information, the system produces a detailed measurement of stressed and unstressed syllables at all acoustic-phonetic levels both in the master and the student signal. However, such measurements are known to be very hard to obtain in a consistent way (Bagshaw P., 1994; Roach P., 1982): so, rather than dealing with syllables, we deal with syllable-like acoustic segments. By a comparison of the two measures and of the remaining portion of signal a corrective diagnosis is consequently issued.

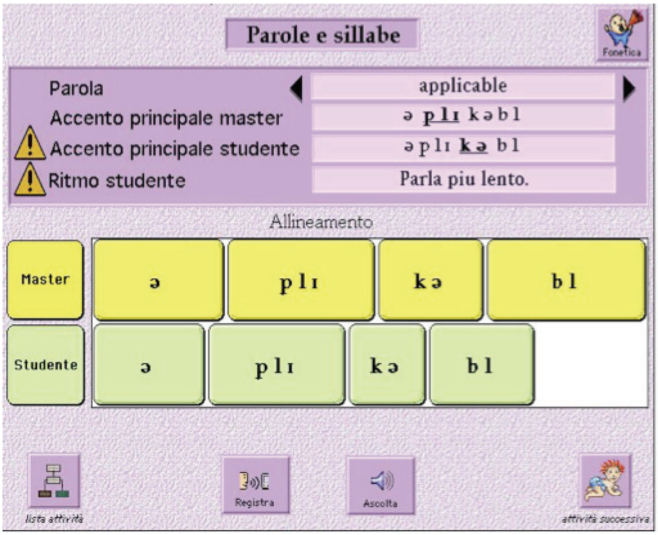
The segmentation and alignment processes can be paraphrased as follows: we have a preprocessing phase in which each word, phonological phrase and utterance is assigned a phonetic description. In turn, the system has a number of restrictions associated to each phone which apply both at subphonemic level, at syllabic level and at word level. This information is used to generate suitable predictions to be superimposed on the segmentation process in order to guide its choices. Both acoustic events and prosodic features are taken into account simultaneously in order to produce the best guess and to ensure the best segmentation.

Each digitalized word, phonological phrase or sentence is automatically segmented and aligned with its phonetic transcript provided by the human tutor, with the following sequence of modules:

- Compute acoustic events for silence detection, silence detection, fricatives detection, noise elimination;
- Extract Cepstral coefficient from the input speech waveform sampled at 16 MHz, every 5 ms for 30 ms frames;
  - Follow a finite-state automaton for phone-like segmentation of speech in terms of phonological features;
- Match predicted phone with actual acoustic data;
- Build syllable-like nuclei and apply further restrictions.

As mentioned above, the student is presented with a master version of an utterance or a word in the language he is currently practising and he is asked to repeat the linguistic item trying to produce a performance as close as possible to the original native speaker version. This is asked in order to promote fluency in that language and to encourage as close as possible mimicry of the master voice.

The item presented orally can be accompanied by situated visual aids that allow the student to objectivize the relevant prosodic patterns he is asked to mimic. The window presented to the student includes three subsections each one devoted to one of the three prosodic features addressed by the system: stressed syllable/syllabic segment - in case of words - or the accented word in case of utterances, intonational curve, overall duration measurement. Word-level exercises (see Figs. 7-8) are basically concentrated on the position of stress and on the duration of syllables, both stressed and unstressed. In particular, Italian speakers tend to apply their word-stress rules to English words, often resulting in a completely wrong performance. They also tend to pronounce unstressed syllables without modifying the presumed phonemic nature of their vocalic nucleus preserving the sound occurring in stressed position: so the use of the reduced schwa-like sound [ə], which is not part of the inventory of phonemes and allophones of the source language, must be learned.



Parole e sillabe

Parola applicable

Accento principale master ə **plɪ** kəbl

Accento principale studente əplɪ **kə** bl

Ritmo studente Parla piu lento.

Allineamento

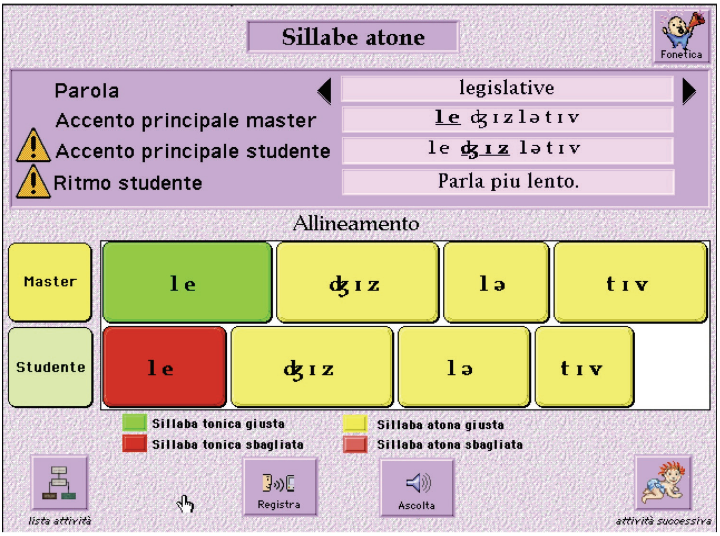
Master ə plɪ kə bl

Studente ə plɪ kə bl

Fig. 7. Syllable Level Prosodic Activities
syll.jpg

The main Activity Window for "Parole e Sillabe"/Words and Syllables is divided into three main sections: in the higher portion of the screen the student is presented with the orthographic and phonetic transcription(in Arpabet) of the word which is spoken aloud by a native speaker's voice. This section of the screen can be activated or disactivated according to which level of Interlingua the student belongs to. We use six different levels (Delmonte R., Cristea D. et al. 1996; Delmonte R., et al. 1996). In particular, the stressed syllable is highlighted between a pair of dots. The main central portion of the screen contains the buttons corresponding to each single syllable which the student may click on. The system then waits for the student performance which is dynamically analysed and compared to the master's. The result is shown in the central section by aligning the student's performance with the master's. According to duration computed for each syllable the result will be a perfect alignment or a misalignment in defect or in excess. Syllables exceeding the master's duration will be shown longer, whereas syllables shorter in duration will show up shorter. The difference in duration will thus be evaluated in proportion as being a certain percentage of the master's duration. This value will be applied to parameters governing the drawing of the related button by HyperCard™. At the same time, in the section below the central one, two warnings will be activated in yellow and red, informing the student that the performance was wrong: prosodic information concerns the placement of word stress on a given syllable, as well as the overall duration (see Bannert 1987; Batliner et al., 1998).

In case of error, the student practicing at word level will hear at first an unpleasant sound which is then followed by the visual indication of the error by means of a red blinking syllable button, the one in which he/she wrongly assigned word stress. This is followed by the rehearsal of the right syllable which always appears in green. A companion exercise takes care of the unstressed portion/s of the word: in this case, the student will focus on unstressed syllables and errors will be highlighted consequently in that/those portion/s of the word. Finally the bottom portion of the window contains buttons for listening and recording on the left, arrows for choosing a new item on the right; at the extreme right side a button to continue with a new Prosodic Activity, and at the extreme left side a button to quit Prosodic Activities.



Sillabe atone

Parola: legislative

Accento principale master: **le** dʒɪzlətɪv

⚠ Accento principale studente: le **dʒɪz** lətɪv

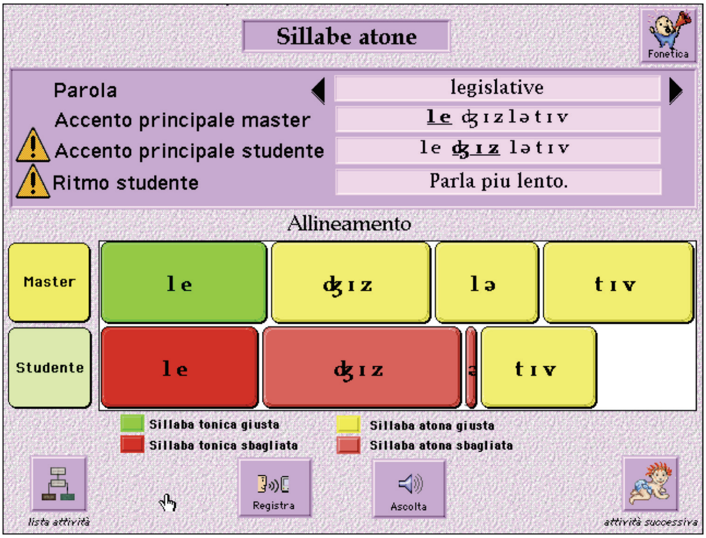
⚠ Ritmo studente: Parla piu lento.

Allineamento

Master	🟩 le	dʒɪz	lə	tɪv
Studente	🟥 le	dʒɪz	lə	tɪv

🟩 Sillaba tonica giusta
🟥 Sillaba tonica sbagliata
🟨 Sillaba atona giusta
🟥 Sillaba atona sbagliata

Fig. 8. Word Stress Prosodic Activities
stress.doc



Sillabe atone

Parola: legislative

Accento principale master: **le** dʒɪzlətɪv

⚠ Accento principale studente: le **dʒɪz** lətɪv

⚠ Ritmo studente: Parla piu lento.

Allineamento

Master	🟩 le	dʒɪz	lə	tɪv
Studente	🟥 le	🟥 dʒɪz	🟥 ə	tɪv

🟩 Sillaba tonica giusta
🟥 Sillaba tonica sbagliata
🟨 Sillaba atona giusta
🟥 Sillaba atona sbagliata

Fig. 9. Unstressed Syllables Prosodic Activities

2.3.3 Phonological rules for phonological phrases

Another important factor in the creation of a timing model of L2 is speaking rate, which may vary from 4 to 7 syllables/sec. Changes in speaking rate exert a complex influence on the durational patterns of a sentence. When speakers slow down, a good fraction of the extra duration goes into pauses. On the other hand, increases in speaking rate are accompanied by phonological and phonetic simplifications as well as differential shortening of vowels and consonants. This usually constitutes another important aspect of English self-learning courseware for syllable-timed L2 speakers. Effects related to speaking rate include compression and elision which take place mainly in unstressed syllables and lead to syllabicity of consonant clusters and of sonorants. As a result of the opposition between weak and strong syllables at word level (Eskénazi, M., 1999), native speakers of English apply an extended number of phonological rules at the level of Phonological Phrase, i.e. within the same syntactic and phonological constituent. These rules may result in syllable deletion, resyllabification and other assimilation and elision phenomena, which are unattested in syllable-timed languages where the identity of the syllable is always preserved word-internally. In rapid/quick colloquial/familiar style of pronunciation in RP of free conversation and dialogue the effects of elision and compression of vowels and consonants can reach 83% elision at word boundary and 17% internal elision (Delmonte R., 2000c).

As far as assimilation is concerned, the main phenomena attested are alveolarization, palatalization, velarization and nasalization some of which are presented here below together with cases attested in our corpus of British English.

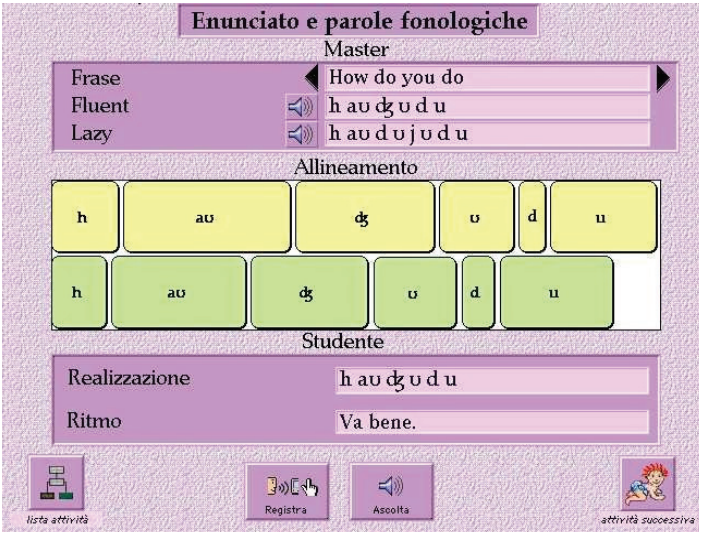
- Homorganic Stop Deletion

The process of homorganic stop deletion is activated whenever a stop is preceded by a nasal or a liquid with the same place of articulation and is followed by another consonant

- In front of voiced/unvoiced fricative
- Homorganic Stop Deletion with Glottalization
- Homorganic Liquid and Voiced Stop Deletion in Consonant Cluster
- Palatalization Rules affect all alveolar obstruents: /t, d, s, z/
- Palatalization of Alveolar Fricative
- Palatalization of Alveolar Nasal
- Palatalization of Alveolar Stop
- Degemination
- Velarization

In order to have Italian students produce fluent speech with phonological rules applied properly we decided to set up a Prosodic Activity which offered the two versions of a single phrase taken from the general course being practised. The student could thus hear both the "lazy" version, with carefully pronounced words, and no rule application taking place; then, the second version, with a fluent and quicker speech is spoken twice. This latter version starts flashing and stops only when the student records his/her version of the phrase.

A comparison then follows which automatically checks whether the student has produced a phrase which is close enough to the "fluent" version. In case the parameters computed are beyond an allowable threshold, the comparison proceeds with the "lazy" version in order to establish how far the student is from the naive pronunciation. The assessment will be used by the Automatic Tutor to decide, together with similar assessments coming from Grammar, Comprehension and Production Activities, the level of Interlingua the student belongs to.



Enunciato e parole fonologiche

Master

Frase: How do you do

Fluent: h aʊ ʤ ʊ d u

Lazy: h aʊ d ʊ j ʊ d u

Allineamento

Studente

Realizzazione: h aʊ ʤ ʊ d u

Ritmo: Va bene.

Fig. 10. Phonological Phrase Level Prosodic Activities

3. Section II: Prosodic tools for self-learning activities in the domain of intonation

3.1 General problems related to intonation in language teaching

In his PhD dissertation and in a number of recent papers M.Jilka (Jilka M. &Möhler, G., 1998; Jilka M., 2000) analyzes the problem of intonational foreign accent (IFA) in the speech of American speakers of German. The definition of what constitutes a case of intonational foreign accent seems fairly straightforward: the intonation in the speech of a non-native speaker must deviate to an extent that is clearly inappropriate for what is considered native. The decision of what intonation is inappropriate or even impossible strongly depends on the surrounding context, much more so than it is the case for deviations in segmental articulation. It is therefore a prerequisite for the analysis of intonational foreign accent that the context be so clear and narrow as to allow a decision with respect to the appropriateness of a particular intonational realization.

This can be done in terms of a categorical description of intonation events based on ToBI labelling. Results show that IFA does indeed include categorical mistakes involving category type and placement, transfer of categories in analogous discourse situations, and deviating phonetic realization of corresponding tonal categories. While such an identification of IFA based on ToBI labeling can be easily achieved in an experimental situation, where transcriptions are all done manually, in a self-learning environment the same results would all be based on the ability of the underlying algorithm to achieve a confident enough comparison between a Master and Student signal. To comply with the idea that only categorical deviations are relevant in the determination of IFA and that it is sensible to propose appropriate corrective feedback only in such cases we need to start from semantically and pragmatically relevant intonational countours as will be discussed in a section below.

As Jilka (2000:Chapter 3) suggests, the main difference in evaluating segmental (allophones) vs suprasegmental (allotones) variations in an L2 student's speech, is that a broader variational range seems to be allowed in the realization of intonational features. We are then faced with the following important assumptions about the significance of variation in the identification of intonational deviations:

- intonation can be highly variable without being perceived as foreign accented (A1)
- context-dependent variation in intonational categories is greater than in segmental categories (A2)

The first assumption (A1) presupposes that the fact that intonation allows a high degree of variation in the choice and distribution of tonal categories is a major aspect aggravating the foreign accent identification process. Noticeable variations may retain the same or a slightly different interpretation, but are not perceived as inappropriate, i.e. foreign-accented. Measurable variations from an assumed prototypical realization may not be perceived at all (thus being basically irrelevant), perceived as different, but not interpreted as such, or actually interpreted as different, but not as foreign.Consequently, a second assumption (A2) about variation in intonation must contend that intonational categories may have more context-dependent different phonetic realizations ("allotones") than segmental categories. This further increases the difficulty in identifying intonational foreign accent, even though, as already mentioned, a number of those additional phonetic realizations do not contribute to foreign accent.

We will compare the two tone inventories as they have been reported in the literature and then we will make general and specific comments on the possibility for an automatic comparing tool to use them effectively. The American English inventory [46], contains five types of pitch accent, two of them monotonal (H*, L*), the other three bitonal (L*+H, L+H*, H+!H*), thus implying an inherent F0 movement (rise or fall) between two targets. Phrasing in American English is determined by two higher-level units, intermediate phrases (ip's) and intonation phrases (IP's). Phrasal tones either high or low in the speaker's pitch range mark the end of these phrases. For intermediate phrases they are called phrase accents (H-, L-), for intonation phrases the term boundary tone (H%, L%) is used. As the terminology suggests, ip's and IP's are ordered hierarchically. An IP consists of one or more ip's and one or more IP's make up an utterance. For this reason, the end of an IP is by definition also the end of an ip, and a boundary tone is always accompanied by a phrase accent, allowing four possible combinations: L-L%, L-H%, H-L% and H-H%.

	American English	Italian
Pitch accents	H*, L*, L+H*, L*+H, H+!H*	H*, L*, L+H*, L*+H, H+L*
Initial Phrasal tones	%H	%H
phrase accents	H-, L-	H-, L-
boundary tones	L-L%, L-H%, H-L%, H-H%	L-L%, L-H%, H-L%, H-H%

Table 1. Tone inventories of American English and Italian

Even though the two inventories are almost identical, the range of variation in intonation contours is used in a much richer way in American English rather than in Italian (Avesani C., 1995).The deviations are summarized in an inventory of nine major differences in the productions of the Dutch speakers (Willems, N., 1983). The listed deviations, which correspond to distinct instances of intonational foreign accent, include what Willems terms:

- the direction of the pitch movement (Dutch speakers may use a rise where British English speakers use a fall)
- the magnitude of the pitch excursion (smaller for the Dutch speakers)
- the incorrect assignment of pitch accents
- differences in the F0 contour associated with specific tonal/phrasal contexts and discourse situations such as continuations (Dutch speakers often produce falls)
- the F0 level at the beginning of an utterance (low in Dutch speakers, but mid in British English speakers) or
- the magnitude of final rises in Yes/No-questions (much greater in Dutch speakers).

Taking into consideration theory-dependent differences in terminology, a number of Willems' results are confirmed in this study's comparison of German and American English.

3.1.1 Teaching intonation as discourse and cultural communicative means

Chun [13.] emphasizes the need to look at research been conducted to expand the scope of intonation study beyond the sentence level and to identify contrasting acoustic intonational features between languages. For example, (Hurley, D. S., 1992) showed how differences in intonation can cause sociocultural misunderstanding. He found that while drops in loudness and pitch are turn-relinquishing signals in English, Arabic speakers of English often use non-native like loudness instead. This could be misinterpreted by English speakers as an effort to hold the floor (ibid. :272-273). Similarly, in a study of politeness with Japanese and English speakers, Loveday (1981) found more sharply defined differences in both absolute pitch and within-utterance pitch variation between males and females in Japanese than between English males and females in English politeness formulas. In addition, the Japanese subjects transferred their lower native language pitch ranges when uttering the English formulas. Low intonation contours are judged by native speakers of English to indicate boredom and detachment, and if male Japanese speakers transfer their low contours from Japanese to English when trying to be polite, this could result in misunderstandings by native English speakers.

As evidence for culture-specificity with regard to the encoding and perception of affective states in intonation contours, Luthy (1983) reported that although a set of "nonlexical intonation signals" (ibid. :19) (associated with expressions like uh-oh or mm-hm in English) were interpreted consistently by a control group of English native speakers, non-native speakers of varied L1 backgrounds tended to misinterpret them more often. He concluded that many foreign students appear to have difficulty understanding the intended meanings of some intonation signals in English because these nuances are not being explicitly taught.Kelm (1987), acknowledging that "correct intonation is a vital part of being understood" (ibid. :627), focused on the different ways of expressing contrastive emphasis in Spanish and English. He investigated acoustically whether the range of pitch of non-native Spanish speakers differed from that of native Spanish speakers. Previous research by Bowen (1975) had found that improper intonation in moments of high emotion might cause a non-native speaker of Spanish to sound angry or disgusted. Kelm found that the native Spanish-speaking group clearly varied in pitch less than the two American groups; that is, native English speakers used pitch and intensity to contrast words in their native language and transferred this intonation when speaking Spanish.

Although the results showed a difference between native and non-native Spanish intonation in contrasts, they did not show the degree to which those differences affect or interfere with communication.

In intonation teaching, one focus has traditionally been contrasting the typical patterns of different sentence types. Pitch-tracking software can certainly be used to teach these basic intonation contours, but for the future, in accordance with the current emphasis on communicative and sociocultural competence, more attention should be paid to discourse-level communication and to cross-cultural differences in pitch patterns. According to Chun (1998), software programs must have the capability to:

- Distinguish the meaningful intonational features with regard to four aspects of pitch change: (a) direction of pitch change (rise, fall, or level), (b) range of pitch change (difference between high and low levels), (c) speed of pitch change (how abruptly or gradually the change happens), and (d) place of pitch change (which syllable(s) in an utterance)
- Go beyond the sentence level and address the multiple levels of communicative competence: grammatical, attitudinal, discourse, and sociolinguistic.

3.2 Intonation practice and visualizatio: our approach

As to Intonational Group detection and feedback, from a number of studies in Dialog Acts it seems clear that intonation is very important in the development of DA classifiers and automatic detector for conversational speech. From the work published in (Shriberg E., et al., 1998) however, we may assume that in the 42 different DA classified only 2 acoustic features were actually considered relevant for the discrimination task: duration and FØ curve. This same type of information is used by our system for intonation teaching. We also assume that word accent is accompanied by *FØ* movement so that in order to properly locate pitch accent we compute *FØ* trajectories first. Then we produce a piecewise stylization which appears in the appropriate window section and is closely followed by the *FØ* trajectory related to the student's performance so that the student can work both at an auditory and at a visual level.

The stylization of an *FØ* contour aims at removing the microprosodic component of the contour. Prosodic representation is determined after *FØ* has been resolved, since *FØ* acts as the most important acoustic correlate of accent and of the intonational contour of an utterance. Basically, to represent the intonational contour, two steps are executed: reducing errors resulting from automatic pitch detection and then stylisation of *FØ* contour. The stylisation of *FØ* contour results in a sequence of segments, very closed to local movements in speaker's intonation. We tackled these problems in a number of papers (see Delmonte R. 1983, 1985, 1987, 1988) where we discuss the relation existing between English and Italian intonational systems both from a theoretical point of view and on the basis of experimental work.

3.2.1 Intonational curve representation

In the generation of an acoustic-phonetic representation of prosodic aspects of speech for computer aided pronunciation teaching, the stylization of an FØ contour aims to remove the microprosodic component of the contour. Prosodic representation is determined after the fundamental frequency has been resolved, since fundamental frequency acts as the most important acoustic correlate of accent and of the intonational contour of an utterance. Basically, to represent the intonational contour, two steps are executed: reducing errors resulting from automatic pitch detection and then stylization of FØ contour. The stylization of FØ contour results in a sequence of segments, very closed to local movements in speaker's intonation. As highlighted above, the pitch resulted is a "direct-period" mirroring. To compute FØ, one might implement the frequency function FØ(t) = 1/T(t). However, by this method dissimmetries will eventually result: on rising portions of T(t), FØ(t) is normally compressed, while on falling portions of T(t), FØ(t) is stretched. As the displayed pitch is intended to put in evidence the rising portions of FØ(t) where accent appears, we prefer to simply compute a symmetric function of the T(t) slope instead of calculating the FØ(t) as 1/T(t). In this way we achieve two goals at one time: the normal compression is thus eliminated, and we save computation time [22.] Delmonte 2010. To classify pitch movements we use four tone types: rising, sharp rising, falling and sharp falling, where the "sharp" versions coincide in fact with main sentence accent and should be time aligned with it. The classification is based on the computation of the distance to the line between beginning and the end of a section, compared on the basis of an a priori established threshold.

3.3 State of the art in prosodic CALL tools: intonation

As we did in the previous section, we report here below a select choice of commercial products and prototypes documented in the literature as being concerned with self-learning tools in the field of prosody, in particular tackling the problem of intonation. In some cases, the same product presented in the previous section reappears here, without repeating the comment, though.

3.3.1 Visi-Pitch visualization

One of the first examples of a program that displays visual pitch curves is a product from Kay Elemetrics called Visi-Pitch that has been available for a number of years for DOS-based personal computers (PCs). With Visi-Pitch, students are able to see both a native speaker's and their own pitch curve simultaneously. Students first speak a sentence into a microphone; their utterance is then digitized and pitch-tracked, and they can see a display of their pitch curve directly under a native speaker's pitch curve of the same sentence. Fig. 11 from Fischer (1986) shows the pitch contours of the French question Qu'est-ce qu'il fait? (What is he doing?) as spoken by a native speaker in the top half, and the same question produced by an American learner in the bottom half.

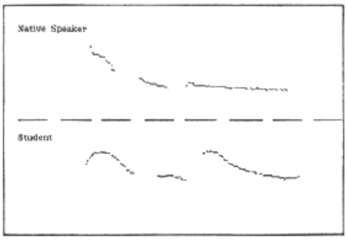


Fig. 11. Visi-Pitch Visualization of Pitch Curve

3.3.2 Auralog's TeLL me More

With the launch of the new version of **TeLL me More** in 2000 (see the website at URL7), Auralog allowed consumers to have easy access to resources that would enhance their language learning.As well as the scoring system, the software also allows the student to accurately visualise not only **pronunciation, but also intonation**. Two types of display mode (waveform and pitch curve) are provided. The student can display them at the same time, or individually. The **waveform** indicates the amplitude of the voice as a function of time **(the notion of energy)**. It represents the sound intensity of the voice and gives a view of the structure of the pronunciation. The **pitch curve** represents frequency variations in the voice. In tandem with the waveform, this curve enables students to make precise comparisons of his or her own intonation with that of the model **(high-pitched/deep)**. This unique display mode is an innovation developed by **Auralog**. **Auralog** is the only software publisher to offer applications which evaluate pronunciation and intonation of both complete sentences and words, and which allows them to be visualised.

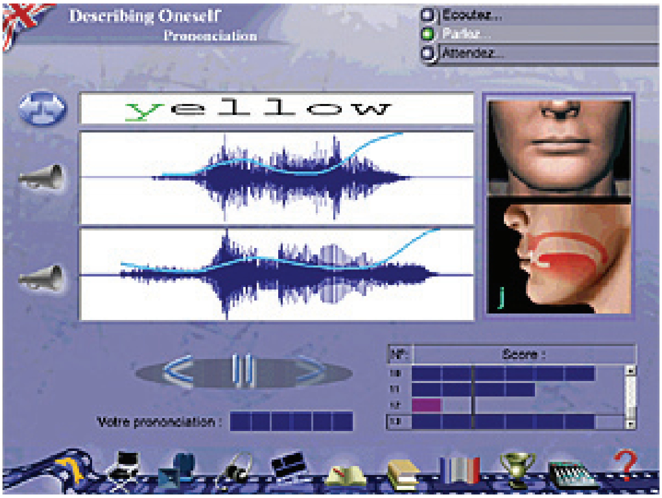


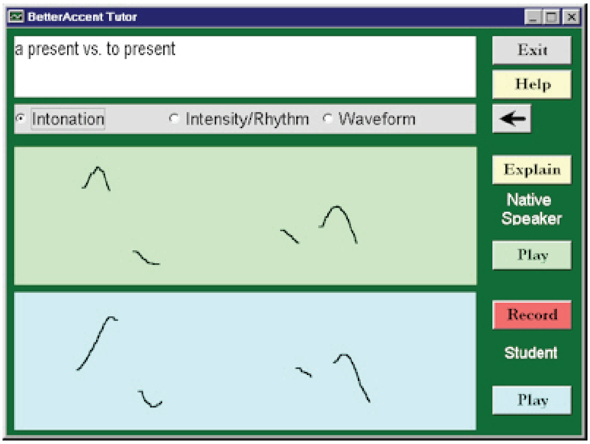
Fig. 12. Auralog's Visualization of Pitch Curve on top of waveform

3.3.3 BetterAccent tutor for english

We repropose here below the visualization of the minimal pair "a present"/"to present"(Fig. 13) where however pitch is used to mark differences between the two phrases. Notice that also the explanation which accompanies the exercise uses information related to intonational curve (Fig. 13.1). The presentation of an utterance is carried out along the same lines: "He said what" (Fig. 14; 14.1). The utterance is an echo question which requires a steep rising tone in coincidence with the wh- word which has been positioned in situ.

3.4 Self-learning activities in the prosodic module: utterance level exercises

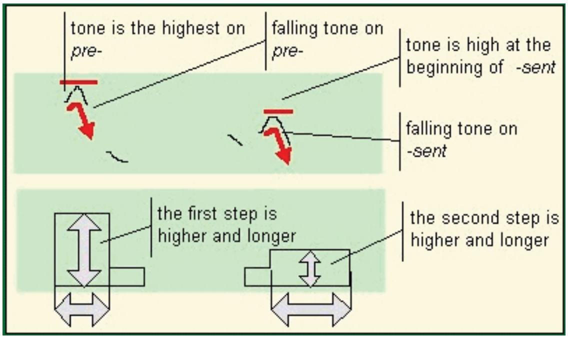
In Utterance Level Prosodic Activities the student is presented with one of the utterances chosen from the course he is following. Rather than concentrating on types of intonation contours in the two languages where performance-related differences might result in remarkable intraspeaker variations, we decided to adopt a different perspective.



a present vs. to present

● Intonation ○ Intensity/Rhythm ○ Waveform

Fig. 13. BetterAccent Visualization of stylized word stress example



tone is the highest on *pre-*

falling tone on *pre-*

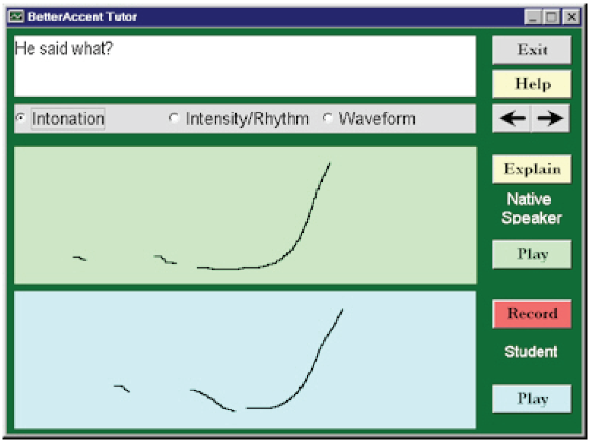
tone is high at the beginning of *-sent*

falling tone on *-sent*

the first step is higher and longer

the second step is higher and longer

Fig. 13.1 BetterAccent evaluation of word stress example



He said what?

● Intonation ○ Intensity/Rhythm ○ Waveform

Fig. 14. BetterAccent Visualization of stylized utterance example

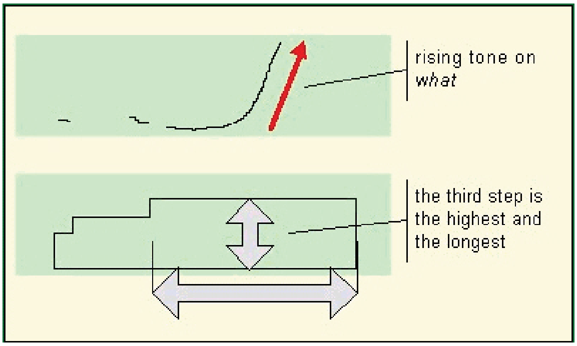


Fig. 14.1 BetterAccent evaluation of utterance example

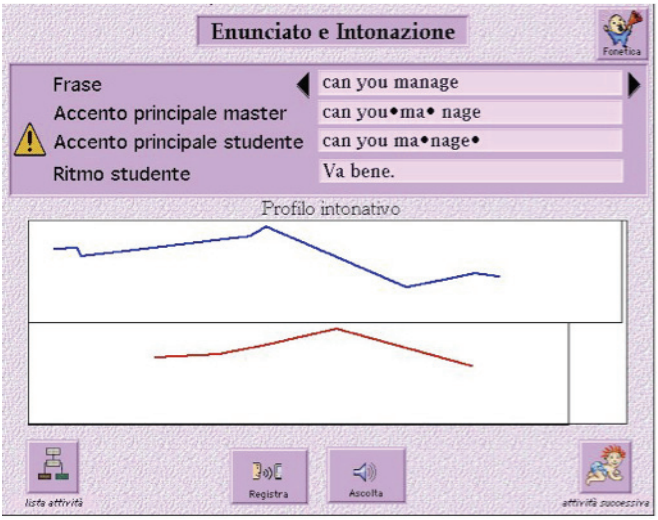
Our approach is basically communicative and focuses on a restricted number of communicative functions from the ones the student is practising in the course he is following (for a different approach see 41 on Japanese-English). Contrastive differences are thus related to pragmatic as well as performance factors. In the course, the student will address some or all of the following communicative functions:

1. Describing actions: habitual, future, current, past; 2. Information: ask for, indicate something/someone, denoting existence/non existence; 3. Socializing: introduce oneself; on the phone; 4. Expressing Agreement and Disagreement; 5. Concession; 6. Rational enquiry and exposition; 7. Personal emotions: Positive, Negative; 8. Emotional relations: Greetings, Sympathy, Gratitude, Flattery, Hostility, Satisfaction; 9. Categories of Modal Meaning, Scales of certainty: i.. Impersonalized: Affirmation, Certainty, Probability, Possibility, Negative Certainty; ii. Personalized: Conviction, Conjecture, Doubt, Disbelief; iii. Scale of commitment; iv. Intentionality; v. Obligation; 10. Mental Attitudes: Evaluation; Veridiction; Committal; Release; Approval; Disapproval; Persuasion; Inducement; Compulsion; Prediction; Tolerance.

All these communicative functions may be given a compact organization within the six following more general functions or macrofunctions:

- ASK; GIVE, OFFER, CONSENT; DESCRIBE; INFORM; SOCIALIZE; ASSERT, SAY, REPLY; EXPRESS EMOTIONS, MODALITIES; MENTAL ATTITUDES.

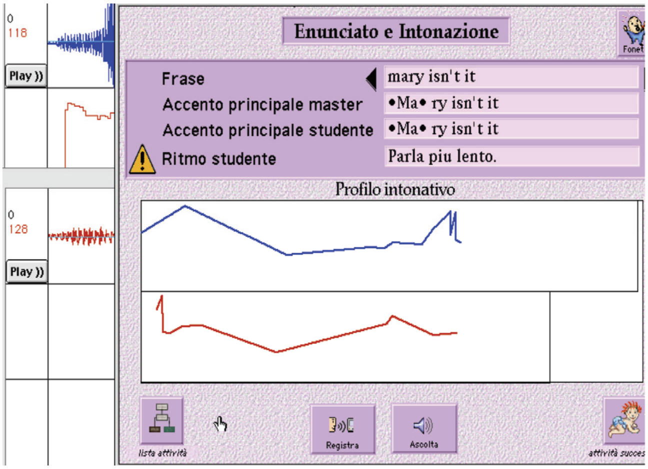
Each function has been given a grading according to a scale of six levels. The same applies to the grading of grammatical items, be they syntactic or semantic, by classifying each utterance accordingly. The level index is used by the Automatic Tutor which has to propose the adequate type of exercise to each individual student (Delmonte R., Cristea D. et al. 1996; Delmonte R. et al., 1996). As far as the Activity Window is concerned - "Enunciato e Intonazione"/Utterance and Intonation, the main difference from Word Level Prosodic Activities discussed above concerns the central main portion of the screen where, rather than a sequence of syllable buttons, the stylized utterance contours appear in two different colours: red for student, blue for master. After each student's rehearsal, the alignment will produce a redrawing of the two contours with different sizes in proportion with the master's one. In the example shown in Fig. 21 below, sentence accent goes on first syllable of the verb “manage” in the Master version, while the student version has accent on the second syllable of the same word “manage”.


Enunciato e Intonazione

Frase	can you manage
Accento principale master	can you•ma• nage
Accento principale studente	can you ma•nage•
Ritmo studente	Va bene.

Fig. 15. Utterance Level Prosodic Activities: 1
fig15.jpg

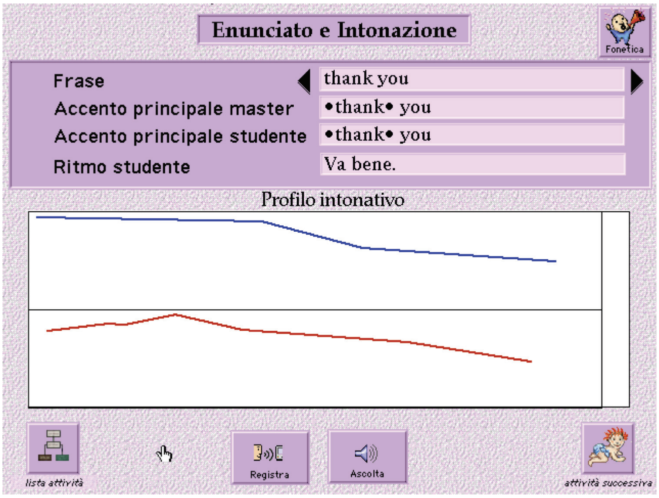
In the second example, we show a Tag-Question, where the difference between the two performances are only in rhythm. Both the initial accent on "Mary" and the final rising pitch on "it" are judged satisfactory by the system which can be seen on the back of the student's activity window.


Enunciato e Intonazione

Frase	mary isn't it
Accento principale master	•Ma• ry isn't it
Accento principale studente	•Ma• ry isn't it
Ritmo studente	Parla piu lento.

Fig. 16. Utterance Level Prosodic Activities: 2
fig16.doc

The third and final example is a simple utterance "Thank you", which however exhibits a big FØ range from the high level of the first peak on the word "Thank" to the low level of the word "you", making it particularly hard for Italian speakers to reproduce it correctly.



Enunciato e Intonazione

Frase: thank you

Accento principale master: •thank• you

Accento principale studente: •thank• you

Ritmo studente: Va bene.

Fig. 17. Utterance Level Prosodic Activities: 3

4. Two systems with animated tutors

Eventually we present two systems that use animated agents or characters to provide feedback to the student and also to guide their activities. The first one has been produced at the Swedish Center for Speech Technology (CTT) at KTH and the second is the result of research work of more than one center, the CSLR.

The Swedish system is called VILLE and is a virtual tutor for Swedish language learners that uses knowledge of phonetics/phonology to help students learn pronunciation (see Engwall and Balter, 2007; Wik et al., 2009). As the authors comment, "the use of embodied conversational agents (ECAs) in computer assisted language learning (CALL) is seen as one way to address feedback issues. Ville guides, encourages and gives corrective feedback to students who wish to develop or improve their Swedish language skills. A first version of Ville was offered in the fall of 2008 to all foreign students at KTH who wanted to learn Swedish. The first version focused on helping students with vocabulary training, providing a model pronunciation of new words and drilling students in memorization exercises… The most serious errors with respect to intelligibility were found to be: lexical stress (insufficient stress marking, or stress on the wrong syllable), consonant deletion in a cluster before a stressed vowel, vowel insertion (epenthesis) in, or before a consonant cluster, vowel and consonant duration errors, vowel quality (difficulties with Swedish vowels not present in L1), and prosodic errors."

The animated tutor has been expanded in its abilities to offer feedback for addressing prosodic errors, in particular in the perception exercises. The result of the implementation of the new 8 Ville capabilities has been studied by means of a questionnaire and students have shown not to care too much to suggestions coming from Ville. In fact, only less skilled students seemed to take advantage of it.

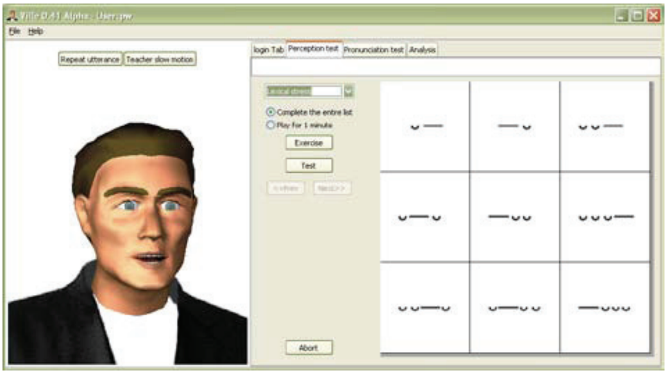


Fig. 18. VILLE animated tutor giving feedback on prosodic exercises

The second system we will comment on is ILT (Italian Literacy Tutor). ILT is a fully comprehensive system for language tutoring expressedly realized for children, the Colorado Literacy Tutor and its companion the Italian Literacy Tutor. Interactive Books, such as that illustrated in Fig. 25 below, incorporate leading edge speech recognition and generation technology, natural language processing tools, computer vision and character animation technologies which provide engaging and immersive learning experiences.The Italian Literacy Tutor is the Italian counterpart of the "Colorado Literacy Tutor" (CLT), a project developed at CSLR (Center for Spoken Language Research, Colorado University Boulder) for English and currently in use in American schools. As its English companion, the ILT integrates two sets of literacy tools, the first one based on speech and animation technology, and the second based on language comprehension technology. These programs are critically useful for children with special needs, in the following four populations: 1) students with reading disabilities, 2) foreign-speaking students with limited Italian/English language proficiency, 3) students with autism spectrum disorder, and 4) students with hearing impairments.

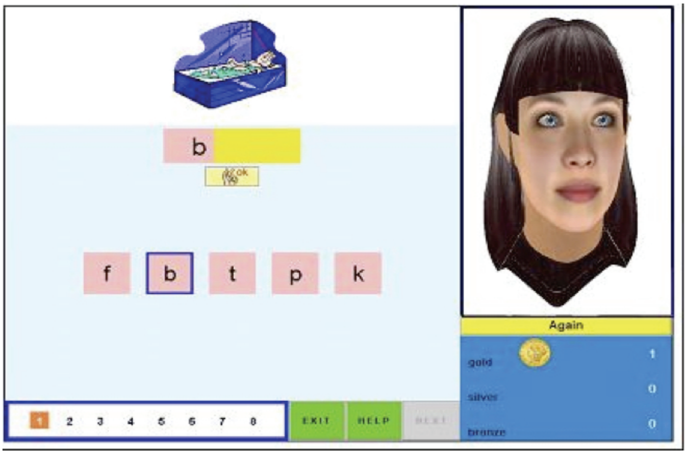


Fig. 19. A phonological and linguistic interactive exercise with an animated agent (LUCIA)

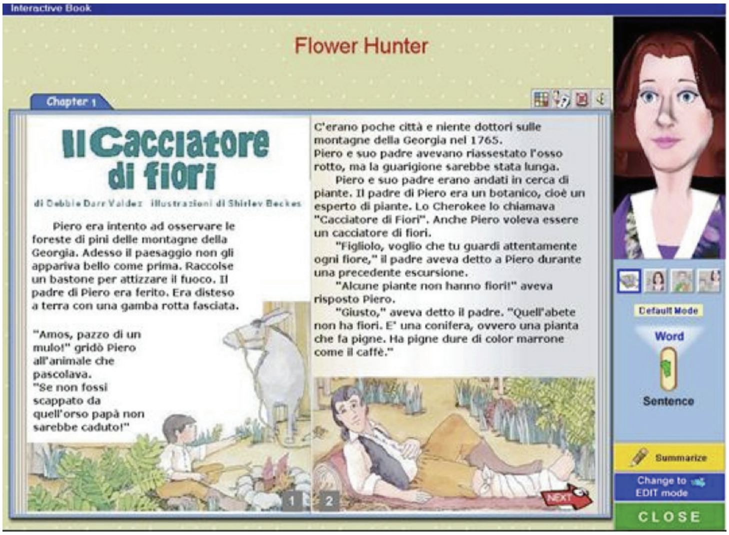
Tutors follow a default sequence from phonological awareness and decoding and encoding of simple consonant-vowel-consonant (CVC) words to more complex orthographic patterns into multisyllable words. Tutors are divided in fact into:

- Phonological Awareness (word, syllable, rhyme, phonemes), with all practice identifying, matching, blending, segmenting, and manipulating these units of spoken language. (see Fig. 19)

Alphabet and Letter-Sound Knowledge ; Reading of Regular Words, from CVC to complex words; Spelling of Regular Words; Reading Sight Words; Spelling Sight Words; Vocabulary

- Comprehension strategies, come into place whenever the children are not successful with the comprehension support and practice within the Books. Word reading, vocabulary, fluency, and comprehension are taught and practiced in Interactive Books, which also assess needs and assign Tutors based on those needs.

Reading Comprehension activites contemplates two types of exercises which requires NLP tools to be used: the first activity is Question/Answering on the contents of the text just read; the other activity, which is more complex to evaluate, is Summarization again of the text just read, which however is no longer visible to the student. In this case, the system activates a Summary evaluation tool which analyses the student text and compares it to a version of the chapter or long paragraph read in a semantic format called Discourse Model (see Delmonte R. 2004, 2007, 2009).



Il Cacciatore di fiori

di Debbie Darr Valdez illustrazioni di Shirley Beckes

Piero era intento ad osservare le foreste di pini delle montagne della Georgia. Adesso il paesaggio non gli appariva bello come prima. Raccolse un bastone per attizzare il fuoco. Il padre di Piero era ferito. Era disteso a terra con una gamba rotta fasciata.

"Amos, pazzo di un mulo!" gridò Piero all'animale che pascolava.
"Se non fossi scappato da quell'orso papà non sarebbe caduto!"

C'erano poche città e niente dottori sulle montagne della Georgia nel 1765.
Piero e suo padre avevano riassestato l'osso rotto, ma la guarigione sarebbe stata lunga.

Piero e suo padre erano andati in cerca di piante. Il padre di Piero era un botanico, cioè un esperto di piante. Lo Cherokee lo chiamava "Cacciatore di Fiori". Anche Piero voleva essere un cacciatore di fiori.

"Figliolo, voglio che tu guardi attentamente ogni fiore," il padre aveva detto a Piero durante una precedente escursione.

"Alcune piante non hanno fiori!" aveva risposto Piero.

"Giusto," aveva detto il padre. "Quell'abete non ha fiori. E' una conifera, ovvero una pianta che fa pigne. Ha pigne dure di color marrone come il caffè."

Fig. 20. An Interactive Book of the Italian version of the CLT with Animated Agent

4.1 Animated speech

Three dimensional animated computer characters associate production of natural or synthetic speech, to a wide variety of facial expressions and emotions, and natural body movements.

The characters' heads can be rotated and made semi or fully transparent, so children can watch how sounds are made to improve their own speech clarity and to detect errors. If a child has, for instance, left out the "l" in spelling "sled," the coach can direct him to watch the tongue movement right after the /s/ in “sled” to discover the missing sound. Children can also compare video capture of their own mouths, in speaking a sound or a word, to the articulation of the coach's mouth. This encourages active and clear speech in the exercises, to improve both the clarity of the child's speech and the underlying precision of his phonological representations for words. They can narrate the book or engage the user in conversational interaction or dialogues to train and test comprehension.

In addition to producing accurate visible speech with associated facial expressions and gestures, animated characters can provide visual feedback to students during learning and conversational interaction. The character can also provide visual feedback and reinforcement, in the form of a head nod, smile, “thumbs up” or other gestures when the student provides correct answers; or look puzzled if the system does not recognize what the student is saying (Cosi et al., 2004a; Cosi et al.,2004b).

4.2 Conclusions and future directions

From what we have shown above, it is possible to make a number of concluding remarks and observations. From what we have shown, it is possible to safely draw a positive conclusion on the introduction of speech technologies in language learning tools. We have also shown that the use of speech technologies is by itself very fruitful in language learning environments but must be complemented by a whole lot of sophisticated tools which take care of pedagogical issues involved in any learning scenarios. In addition to that, speech technologies require empirical research to properly assess the adequateness of its architecture and curriculum for the intended domain and pedagogical objectives, which do not coincide directly with human directed teaching activities. It is still hard to think in terms of linguistic issues when providing feedback to students: as we saw, only the identity of the sounds or syllable or word involved in speech recognition can be addressed by feedback in currently available ASR. As to prosodic issues, only a few of the problems involved in prosodic learning can be detected and properly addressed when producing feedback. So there is still a long way to go to teach using CALL systems (Delmonte R. 2002b, 2003a).

The most challenging scenario is certainlyrepresented by the system at the end of the paper, where animated tutors are incorporated in a full-fledged system for literacy tutoring for children or for the teaching of pronunciation.Animated characters incorporate the technology of the future of language tutoring and constitute the test-bed for interactive activities where both speech synthesis and recognition are used and require implementation of modules for emotional speech. Here we would like to go back to the statements reported at the beginning of this chapter, by Sproat and van Santen, where the complexity of the task facing the use of speech technology is clearly outlined and covers the whole set of scientific domains associated to human language sciences. Animated tutors will certainly become a realilty in a near future, but a lot of work is still needed to address emotional issues both in the visual and in the speech domain.

5. References

[1] Avesani, C. (1995). ToBIT: un sistema di trascrizione per l’intonazione italiana, In: *Atti delle 5^ Giornate di Studio GFS*, Povo(TN), pp. 85-98.

[2] Bagshaw, P. (1994). *Automatic Prosodic Analysis for Computer Aided Pronunciation Teaching*, Unpublished PhD Dissertation, Univ. of Edinburgh, UK.

[3] Bagshaw, P., Hiller, S., & Jack, M. (1993). Computer aided intonation teaching, In: *Proceedings of Eurospeech*, pp. 1003-1006.

[4] Bannert, R. (1987). From Prominent Syllables to a Skeleton of Meaning: A Model of a Prosodically Guided Speech Recognition, In *Proceedings of the XIth ICPhS*, Vol.2, 22.4.

[5] Batliner, A., R.Kompe, A.Kiessling, M.Mast, H.Niemann, NoethE. (1998). M - Syntax + Prosody: A syntactic-prosodic labelling scheme for large spontaneous speech databases, in Speech Communication, Vol. 25, No. 4, pp. 193-222.

[6] Bernstein, J. (1998). New uses for speech technology in language education, In: *Proceedings ISCA Workshop on Speech Technology in Language Learning, STiLL 98*, Marhollmen, pp. 173-176.

[7] Bernstein, J. (1999), "PhonePass testing: Structure and construct," Ordinate Corporation, Menlo Park, CA May 1999.

[8] Bernstein, J., & Franco, H. (1995). Speech recognition by computer. In: N. Lass (Ed.), *Principles of experimental phonetics*, New York: Mosby, pp. 408-434.

[9] Bertinetto, P.M. (1980). The Perception of Stress by Italian Speakers, *Journal of Phonetics*, Vol. 8, pp. 385-395.

[10] Bowen, J. D. (1975). *Patterns of English pronunciation*. New York: Newbury House.

[11] Campbell, W. (1993). Predicting Segmental Durations for Accomodation within a Syllable-Level Timing Framework, In: *Proc. Eurospeech '93*, pp. 1081-1085.

[12] Campbell, W., IsardS. (1991). Segment durations in a syllable frame, In: *Journal of Phonetics*, Vol. 19, pp. 37-47.

[13] Chun, D.M. (1998). "Signal Analysis Software For Teaching Discourse Intonation", LLTJ, Language Learning & Technology, Vol. 2, No. 1, pp. 61-77.

[14] Cosi, P., R. Delmonte, S. Biscetti, R. A. Cole, B. Pellom, van VurenS. (2004). ITALIAN LITERACY TUTOR: tools and technologies for individuals with cognitive disabilities, in R.Delmonte & S.Tonelli(eds), In: *Proc.INSTIL/ICALL2004*, Venezia, pp. 207-215.

[15] Cosi, P., R. Delmonte, S. Biscetti, ColeR. A. (2004). ITALIAN LITERACY TUTOR: un adattamento all'italiano del "Colorado Literacy Tutor", A. Andronico, P. Frignani, G. Poletti (a cura di), *Atti DIDAMATICA 2004*, Ferrara, pp. 249-253.

[16] Cucchiarini, C., H. Strik, and BovesL. (1997a). Using speech recognition technology to assess foreign speakers' pronunciation of Dutch, *In: Proc. Third international symposium on the acquisition of second language speech: NEW SOUNDS 97*, Klagenfurt, Austria.

[17] Cucchiarini, C., S. Strik, and BovesL. (1997b). Automatic evaluation of Dutch pronunciation by using speech recognition technology, In: Proc. *IEEE Automatic Speech Recognition and Understanding Workshop*, Santa Barbara, CA, 1997.

[18] Delmonte, R. (1983). A Phonological Processor for Italian, *Proceedings of the 2nd Conference of the European Chapter of ACL*, Pisa, pp. 26-34.

[19] Delmonte, R. (1985). Parsing Difficulties & Phonological Processing in Italian, *Proceedings of the 2nd Conference of the European Chapter of ACL*, Geneva, pp. 136-145.

[20] Delmonte, R. (1987). The Realization of Semantic Focus and Language Modeling, In:*Proceeding of the International Congress of Phonetic Sciences*, Tallinn (URSS), pp. 100-104.

[21] Delmonte, R. (1988). Focus and the Semantic Component, In:*Rivista di Grammatica Generativa*, pp. 81-121.

[22] Delmonte R., M. Petrea, C. Bacalu (1997). *SLIM Prosodic Module for Learning Activities in a Foreign Language*, Proc.ESCA, *Eurospeech'97*, Rhodes, Vol.2, pp.669-672.

[23] Delmonte, R. (1999). Prosodic Variability: from Syllables to Syntax through Phonology, in *Atti IX Convegno GFS-AIA*, Venezia, pp. 133-146.

[24] Delmonte, R. (2000). SLIM Prosodic Automatic Tools for Self-Learning Instruction, *Speech Communication*, Vol. 30, pp. 145-166.

[25] Delmonte R.(2002). Feedback generation and linguistic knowledge in 'SLIM' automatic tutor, *ReCALL*, Vol. 14, No. 1, Cambridge University Press, pp. 209-234.

[26] Delmonte R.(2003). Linguistic Knowledge and Reasoning for Error Diagnosis and Feedback Generation, In: Trude Heift and Mathias Schulze(eds.), Error Analysis and Error Correction in Computer-Assisted Language Learning, *CALICO Spring 2003 special issue, CALICO JOURNAL*, Southwest Texas State University, pp.513-532.

[27] Delmonte, R. (2004). Evaluating Students' Summaries with GETARUNS, *Proc.INSTIL/ICALL2004*, Unipress, Padova, pp. 91-98.

[28] Delmonte R., (2007), *Computational Linguistic Text Processing – Logical Form, Semantic Interpretation, Discourse Relations and Question Answering*, Nova Science Publishers, New York.

[29] Delmonte R., 2009. *Computational Linguistic Text Processing – Lexicon, Grammar, Parsing and Anaphora Resolution*, Nova Science Publishers, New York.

[30] Delmonte, R. (2010). Prosodic tools for language learning, *International Journal of Speech Technology*, Vol. 12, No. 4, pp.161–184.

[31] Delmonte, R., Andrea Cacco, Luisella Romeo, Monica Dan, Max Mangilli-Climpson, Stiffoni F.(1996). SLIM - A Model for Automatic Tutoring of Language Skills, *Ed-Media 96, AACE*, Boston, pp. 326-333.

[32] Delmonte, R., Dan Cristea, Mirela Petrea, Ciprian Bacalu, Stiffoni F. (1996). Modelli Fonetici e Prosodici per SLIM, *Atti 6° Convegno GFS-AIA*, Roma, pp. 47-58.

[33] Engwall O. and Balter O. (2007). Pronunciation feedback from real and virtual language teachers," *Computer Assisted Language Learning*, vol. 20, pp. 235-262.

[34] Eskénazi, M. (2009). An overview of spoken language technology for education, *JSC (Journal of Speech Communication)*, Vol. 51, No. 10.

[35] Eskénazi, M. (1999). "Using Automatic Speech Processing for Foreign Language Pronunciation Tutoring: Some Issues and a Prototype", *Language Learning & Technology*, Vol. 2, No. 2, pp. 62-76.

[36] Eskénazi, M., Yan Ke, Jordi Albornoz, Probst, K. (2000). Update on the Fluency Pronunciation Trainer, Dundee, Scotland, pp. 73-76.

[37] Eskénazi, Maxine, Angela Kennedy, Carlton Ketchum, Robert Olszewski, and Pelton, G.(2007). The NativeAccentTM pronunciation tutor: measuring success in the real world, in *Proc. SLaTE 2007.*

[38] Fischer, L. B. (1986). *The use of audio/visual aids in the teaching and learning of French*. Pine Brook, NJ: Kay Elemetrics Corporation.

[39] Franco, H., Victor Abrash, Kristin Precoda, Harry Bratt, Ramana Rao, and Butzberger, J. (2000). The SRI EduSpeak System: Recognition and Pronunciation Scoring for Language Learning, In: *Proc. InSTiLL*, Dundee, Scotland, pp.123-128.

[40] Grover, C., J. Fackrell, H. Vereecken, J.-P. Martens, Van Coile, B. (1998). Designing Prosodic Databases for Automatic Modelling in 6 Languages, *Proceedings of ESCA/COCOSDA Workshop on Speech Synthesis*, Australia, pp. 93-98.

[41] Havranek, G., "When is corrective feedback most likely to succeed?, (2002).*International Journal of Educational Research*, vol. 37, pp. 255-270.

[42] Hiller, S., E.Rooney, J.Laver and Jack M. (1993). SPELL: An automated system for computer-aided pronunciation teaching, *Speech Communication*, Vol. 13, pp. 463-473.

[43] Hurley, D. S. (1992). Issues in teaching pragmatics, prosody, and non-verbal communication. *Applied Linguistics*, Vol. 13, No. 3, pp. 259-281.

[44] Jilka, M. & Möhler, G. (1998). Intonational Foreign Accent: Speech Technology and Foreign Language Teaching. In: *Proceedings of the ESCA Workshop on Speech Technology in Language Learning*, Marholmen, pp. 115 – 118

[45] Jilka, M., (2000). The Contribution of Intonation to the Perception of Foreign Accent. Doctoral Dissertation, Arbeiten des Instituts für Maschinelle Sprachverarbeitung (AIMS) Vol. 6, No. 3, University of Stuttgart.

[46] Jilka, M., PhD. Dissertation, available at http://www.ims.uni-stuttg art/phonetik/matthias/

[47] Kanters, Sandra, Catia Cucchiarini, and Strik, H. (2009). The Goodness of Pronunciation Algorithm: a Detailed Performance Study, In:*Proc. SLaTE 2009*.

[48] Kawai, G., Hirose K. (1997). A Call System using Speech Recognition to Train the Pronunciation of Japanese Long Vowels, the Mora Nasal and Mora Obstruent, In:*Proc. Eurospeech97*, Vol.2, pp. 657-660.

[49] Kelm, O. R. (1987). An acoustic study on the differences of contrastive emphasis between native and non-native Spanish speakers. *Hispania*, No. 70, pp. 627-633.

[50] Kim, Y., H.Franco, Neumeyer L. (1997). Automatic Pronunciation Scoring of Specific Phone Segments for Language Instruction, in *Proc. Eurospeech97*, Vol.2, pp. 645-648.

[51] Klatt, D. (1987). Review of text-to-speech conversion for English, *J.A.S.A,* No. 82, pp. 737-797.

[52] Komissarchik, E., Komissarchik J. (2000a). Application of Knowledge-Based Speech Analysis to Suprasegmental Pronunciation Training, *AVIOS 2000 Proceedings*, San Jose, California.

[53] Komissarchik, E., Komissarchik J. (2000b). BetterAccent Tutor – Analysis and Visualization of Speech Prosody, In: *Proc. Speech Technology in Language Learning*, Dundee, Scotland.

[54] Lehiste, I. (1977). Isochrony reconsidered, In:*Journal of Phonetics*, No. 3, pp. 253-263.

[55] Loveday, L. (1981). Pitch, politeness and sexual role: An exploratory investigation into the pitch correlates of English and Japanese politeness formulae. *Language and Speech*, No. 24, pp. 71-89.

[56] Luthy, M. J. (1983). Nonnative speakers' perceptions of English "nonlexical" intonation signals. *Language Learning*, Vol. 33, No. 1, pp. 19-36.

[57] Meador, J., F.Ehsani, K.Egan, and Stokowski S. (1998). An Interactive Dialog System for Learning Japanese, In:*Proc. STiLL '98*, pp. 65-69.

[58] Neumeyer, L., Franco, H., Abrash, V., Julia, L., Ronen, O., Bratt, H., Bing, J., Digalakis, V., Rypa, M. (1998). "WebGrader™: A multilingual pronunciation practice tool", In:*Proceedings ESCA Workshop on Speech Technology in Language Learning* (*STiLL)' 98*, pp. 61-64.

[59] Price, P. (1998). How can Speech Technology Replicate and Complement Good Language Teachers to Help People Learn Language?, in *Proc. STiLL '98*, pp. 103-106.

[60] Ramus, F. and Mehler J. (1999). Language identification with suprasegmental cues: A study based on speech resynthesis. In: *Journal of the Acoustic Society of America*, Vol. 105, No. 1, pp. 512-521.

[61] Ramus, F., Nespor M., Mehler J.(1999). Correlates of linguistic rhythm in the speech signal, *Cognition*, Vol. 26, No. 2, pp. 145-171.

[62] Roach, P., (1982). On the distinction between stress-timed and syllable-timed languages, In: Linguistic Controversies, D.Crystal (ed.), Edward Arnold, London, pp. 73-79.

[63] Ronen, O., L.Neumeyer, Franco H.(1997). Automatic Detection of Mispronunciation for Language Instruction, in Proc. Eurospeech97, Vol.2, pp. 649-652.

[64] Rooney, E., Hiller, S., Laver, J., & Jack, M. (1992). Prosodic features for automated pronunciation improvement in the SPELL system. *Proceedings of the International Conference on Spoken Language Processing*, Banff, Canada, pp. 413-416.

[65] Shriberg, E., R. Bates, A. Stolcke, P. Taylor, D. Jurafsky, K. Ries, N. Coccaro, R. Martin, M. Meteer, Van Ess-Dykema C. (1998). Can Prosody Aid the Automatic Classification of Dialog Acts in Conversational Speech?, In:*LanguageandSpeech-Special Issue on Prosody and Conversation*, Vol. 41, No. 3-4,pp.439-487.

[66] Umeda, N. (1977). "Consonant Duration in American English", JASA , Vol. 61, pp. 846-58.

[67] URL7=http://www.tellmemore.com

[68] van Santen, J. (1997). Prosodic Modeling in Text-to-Speech Synthesis, In:*Proc. Eurospeech97*, Vol.1, pp. 19-28.

[69] van Santen, J., C.Shih, B.Möbius, E.Tzoukermann, Tanenblatt M. (1997). Multi-lingual durational modeling, In:*Proc. Eurospeech97*, Vol.5, pp. 2651-2654.

[70] van Son, R., van Santen J. (1997). Strong Interaction between Factors Influencing Consonant Duration, In:*Proc. Eurospeech97*, Vol.1, pp. 319-322.

[71] Wik, P., Hincks, R. &Hirschberg J. (2009). Responses to Ville: A virtual language teacher for Swedish, In: *Proc. SLaTE2009*.

[72] Willems, N. (1983). English Intonation from a Dutch Point of View. Doctoral Dissertation, University of Utrecht.

[73] Zechner, K., Derrick Higgins, Xiaoming Xi, (2009). SpeechRater™: A Construct-Driven Approach to Scoring Spontaneous Non-Native Speech, *Proc. SLaTE 2009*.

[74] Zechner, K., I.I. Bejar, and Hemat, R. (2007). Towards an understanding of the role of speech recognition in non-native speech assessment, Educational Testing Service, Princeton, 2007.

Part 3

Language Modeling

N-Grams Model For Polish

Bartosz Ziółko, Dawid Skurzok
Department of Electronics
AGH University of Science and Technology
Kraków, Poland

1. Introduction

N-grams are very popular in automatic speech recognition (ASR) systems (Young et al., 2005), (Lamere et al., 2004), (Whittaker & Woodland, 2003), (Hirsimaki et al., 2009). They have been found as the most effective models for several languages. N-grams calculated by us will be used for the language model of a large vocabulary Polish ASR system and other outside application, first of them being SnapKeys virtual keyboard. Our earlier results and process of collecting statistics were described already (Ziółko, Skurzok & Ziółko, 2010). In this chapter we want to describe a complete model and its applications.

Creating a large vocbulary model of Polish is a difficult task because there are fewer Polish text corpora then for English. What is more, Polish is very inflected in contrast to English. The rich morphology causes difficulties in training language models due to data sparsity. Much more text data must be used for inflected languages than for positional ones to achieve the model of the same efficiency (Whittaker & Woodland, 2003).

2. Available text corpora for Polish

There are 280 000 words in Polish *myspell* dictionary. The number contains only basic forms. With all inflections, over 1 000 000 words can be easily expected. This is just without proper names. In our case we noted several million words, because of proper names and errors.

The IPI PAN Corpus (Przepiórkowski, 2004) is a the main professional and official corpus of Polish texts. Currently, there are over 250 million segments which are morphosyntactically annotated in a publicly available version. It was developed by the Linguistic Engineering Group at the Institute of Computer Science, Polish Academy of Sciences. The same group works on creating much larger corpus of Polish toegether with some publishers.

However, there are several larger corpora of Polish. They are often not annotated and not available publicly. It is a result of a specific approach of Polish law to copyrights. It is legal to download any texts from Internet, even, if they were put there without authors permission. However, it is not legal to upload any such materials anywhere without persmission and this law is very strictly enforced.

This is why natural language researchers working on Polish do not offer their resources both for free or commercialy, eventhough, some of them collected relatively large data sets. For the mentioned reason, it is not easy to estimate real sizes of corpora of Polish texts.

Newspaper articles in Polish were used as our first corpus. They are Rzeczpospolita newspaper articles taken from years 1993-2002. Several millions of Wikipedia articles in Polish

Corpus	MBytes	Mwords	Perplexity
Rzeczpospolita journal	879	104	8 918
Wikipedia	754	97	16 436
Literature	490	68	9 031
Transcripts	325	32	4 374
Literature 2	6500	949	6181
Literature 3	285	181	4258

Table 1. Analysed text corpora with their sizes, perplexity. More data (websites and literature) were already collected but not analysed yet

Corpus	Basic forms	1-grams	2-grams	3-grams
Rzeczpospolita journal	832 732	856 349	18 115 373	43 414 592
Wikipedia	2 084 524	2 623 358	31 139 080	61 865 543
Literature	610 174	1 151 043	23 830 490	50 794 854
Transcripts	183 363	381 166	6 848 729	16 283 781
Literature 2	?	6 162 530	153 152 158	441 284 743
Literature 3	?	1 229 331	36 297 382	93 751 340

Table 2. The number of different *n*-grams

Corpus	single 1-grams	%	1-grams with errors	%
Rzeczpospolita journal	363 391	42.4	7435	0.86
Wikipedia	379 147	46.5	108 338	4
Literature	467 376	41	75 204	6.5
Transcripts	147 440	39	1 373	0.4
Literature 2	3 552 379	57.6	343 211	5.27
Literature 3	485 713	39.5	6040	0.48

Table 3. Errors in the analysed corpora

made another corpus. The smallest articles were removed from the corpus. In this way we avoided some Wikipedia patterns like *Zawada - a village in Poland, located in Łódzki voivodeship, in Tomaszewski powiat, in Tomaszów Mazowiecki parish. During years 1975-1998, the village belonged to piotrkowskie voivodeship.* There are over 50 000 villages described using exactly same pattern. As a result before we removed them, this pattern provided the list of 5 most common 3-grams, even after combining Wikipedia with two other corpora. Several thousands literature books in Polish from different centuries were used. The fourth corpus is a collection of transcripts from the Polish Parliament, its special investigation committees and Solidarność meetings. They contain mainly transcribed speech but also some comments on the situation in rooms. It is not as big as others but the only one containing transcriptions of spoken language. What is more its topics are law oriented, which corresponds very well with our project, which provides ASR system for Police, administration and other governmental institutions. We are still in the process of collecting more Polish text corpora and combining the statistics of the described ones.

In all cases perplexity is very high comparing to typical English corpora. It is because of inflected nature of Polish and significant number of proper names in the corpora.

3. Problems with processing Polish corpora

Some English, Russian, Chinese and other foreign words appeared in the statistics as well as single letters. Such words could be effects of including some foreign quotes in articles. However, most of the foreign words are proper names and they appeared in Polish sentences. After an analysis of results of collecting n-gram statistics from various corpora, we decided that some supervised correction is necessary. Because of the amount of data, the choice of strategy in this process was crucial from financial point of view. We designed and implemented software Fixgram (Ziółko, Skurzok & Michalska, 2010) to optimise n-gram corrections by time efficiency.

The list of words for corrections is preprepared on a server. This is why, it is partly unsupervised method. Three schemes of preparing words were implemented. The first one is finding pairs of words which are different only by orthographic notation, in example *rz* and *ż*. The second is by finding words with any non-Polish letters. The third method is by comparison with *myspell* dictionary. The words which do not exist in *myspell* are also more likely to be errors then others. A user of Fixgram receives a database of words chosen for corrections to save time spent on automatic search for them in a database during human work. All chosen words are given to the Fixgram user in order by the number of times they appeared in a corpus. All, less common cases will be done automatically, typically by deleting. There is no reason in spending human time for rare cases which are likely to be incorrect and not crucial for statistics. The results from one corpus can be transferred to another one. Sometimes human decisions can be generalised and used for less often cases.

A few types of problems were encountered. The first one are Chinese and English proper names. They appeared quite frequently in the newspaper corpus. Often two Chinese names were detected as orthographic errors because of differences only in *ch* and *h*. Chinese proper names tend to be also often in addition to a Polish word, so one orthographic transcription is for a correct Polish word and the other for Chinese proper name.

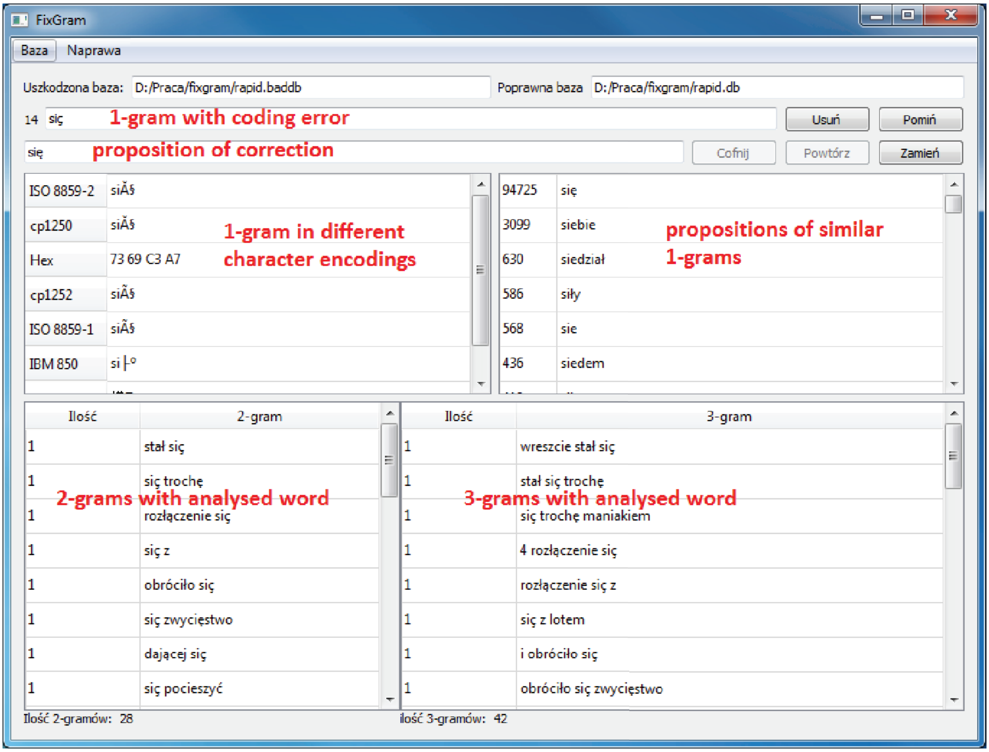
Another type of a problem are words which were split into two words with a space so they appeared as two separate words in n-grams. These are difficult to be found automatically.

There are also words which are wrongly formatted (not in UTF-8). Most of them are not in any of known to us standards for Polish letters. This is because we changed all typical standards to UTF-8 before collecting the statistics. These words can still be recognised by a human, as typically there is only one special Polish letter and other are standard Latin letters.

Fixgram (Fig. 1) (Ziółko, Skurzok & Michalska, 2010) presents contexts of each word (2- and 3-grams). It makes correcting these cases much easier. Apart from that, quite a lot of Russian words and single letters (in Cyrillic) were discovered. All of them were removed.

Several automatically detected words were actually correct. For example, there are plenty of similar surnames with an only difference in Polish special ortographic notation. There were some other words which are correct with both orthographic transcriptions but different senses, like morze (Eng. sea) and może (Eng. maybe). These cases were kept in the n-gram database by a human decision.

We have an extra collection of texts from Internet. However, to ensure proper quality, these websites will be first filtered using statistics collected from literature and journals. Only websites with very little new 1-grams will be accepted and added to the model. This process will be repeated iteratively several times. The decisions can be also taken using phoneme statistics of Polish which we also already calculated and currently are improving. In these ways we want to use Internet resources to analyse as much text as possible, but to avoid including texts of low quality or non-Polish ones.


FixGram
Baza
Naprawa
Usuń
Pomiń
Cofnij
Powtórz
Zamień
1-gram with coding error
proposition of correction
1-gram in different character encodings
propositions of similar 1-grams
2-grams with analysed word
3-grams with analysed word

Uszkodzona baza: D:/Praca/fixgram/rapid.baddb

Poprawna baza D:/Praca/fixgram/rapid.db

14 sіç

się

ISO 8859-2	siĂ§
cp1250	siĂ§
Hex	73 69 C3 A7
cp1252	siÃ§
ISO 8859-1	siÃ§
IBM 850	si├º

94725	się
3099	siebie
630	siedział
586	siły
568	sie
436	siedem

Ilość	2-gram
1	stał sіç
1	sіç trochę
1	rozłączenie sіç
1	sіç z
1	obróciło sіç
1	sіç zwycięstwo
1	dającej sіç
1	sіç pocieszyć

Ilość 2-gramów: 28

Ilość	3-gram
1	wreszcie stał sіç
1	stał sіç trochę
1	sіç trochę maniakiem
1	4 rozłączenie sіç
1	rozłączenie sіç z
1	sіç z lotem
1	i obróciło sіç
1	obróciło sіç zwycięstwo

ilość 3-gramów: 42

Fig. 1. Screenshot of our Fixgram (Ziółko, Skurzok & Michalska, 2010) software to correct n-gram statistics

4. Results

The most common words in Polish are presented in Table 4. Most frequent 2-grams and 3-grams show Tables 5 and 6. Collected statistics show that the amount of text we used was enough to create representative statistics for 1-grams, 2-grams and even for 3-grams. It is the first such model for Polish.

The most popular 1-grams in Polish are mainly pronouns, what is not surprising. The most popular 2- and 3- grams contain often a dot. Its commonality in the statistics is overhelming but the probability that a particular word starts or ends a sentence is indeed much higher then that two exact words appear next to each other.

The English translations were provided in Table 4 with 1-grams. However, it is quite difficult to translate pronouns without a context. This is why, there are sometimes several translations and eventhough, they are only brief and not complex translations. One of the commonly used words is się. It is a reflexive pronoun. It could be translated as oneself, but it is much more common in Polish than in English. It is used always, if a subject activity is conducted on herself or himself.

The distribution of 1-grams is presented in Fig. 2. The histogram has an expected shape, similarly to histograms of 2- and 3-grams.

word (Eng.)	%	word (Eng.)	%	word (Eng.)	%
.	8.235	kiedy (when)	0.160	niego (him)	0.085
i (and)	2.365	gdy (while)	0.157	jako (as)	0.085
w (in)	0.234	by (would)	0.150	lecz (but)	0.083
się (r.p.)	2.255	ten (this)	0.141	gdzie (where)	0.082
nie (no,not)	1.714	ma (has)	0.139	je (them f.,	0.081
na (on, at)	1.635	który (which m.)	0.138	eats)	
z (with)	1.498	jednak (however)	0.132	nich (them)	0.080
do (to,till)	1.093	ją (her)	0.131	nas (us)	0.078
to (it, this)	0.928	pod (under)	0.129	siebie (themselves)	0.078
że (that)	0.890	była (was f.)	0.129	lub (or)	0.078
a (and)	0.690	przed (before,	0.128	aby (so as)	0.077
o (about, at)	0.549	in front of)	-	te (these f.)	0.076
jak (how,like)	0.485	nawet (even)	0.128	tych (these m.)	0.075
jest (is)	0.440	pan (master)	0.126	pani (madam)	0.075
po (after)	0.426	teraz (now)	0.124	niż (than)	0.074
ale (but)	0.396	ja (I)	0.123	ani (neither)	0.074
co (what)	0.393	bardzo (very)	0.122	(f. prop. name)	-
tak (yes)	0.366	przy (next to)	0.121	można (may)	0.071
za (for,	0.343	są (are)	0.119	nigdy (never)	0.069
behind, by)	-	które (which f. pl.)	0.119	właśnie (just)	0.069
od (from, since)	0.319	tu (here)	0.114	sam (alone)	0.068
jego (his)	0.282	być (be)	0.111	były (were f.)	0.067
przez (through)	0.271	więc (so)	0.110	która (which f.)	0.066
jej (her)	0.262	też (also)	0.107	dobrze (well)	0.065
tym (this)	0.258	tej (this f.)	0.106	niej (her)	0.065
go (him)	0.257	on (he)	0.102	także (also)	0.064
już (yet,	0.252	wszystko (all)	0.101	zawsze (always)	0.063
already)	-	tam (there)	0.101	ty (you)	0.061
tylko (only)	0.230	jeśli (if)	0.101	ta (this f.)	0.060
czy (if)	0.223	nim (him)	0.101	domu	0.060
tego (that, hereof)	0.216	coś (something)	0.101	(house gen.)	-
mnie (me)	0.211	będzie (will be)	0.100	albo (or)	0.060
był (was m.)	0.203	bo (because)	0.099	sposób (way,	0.060
było (was n.)	0.200	nic (nothing)	0.098	method)	-
ze (of, by,	0.190	bez (without)	0.097	oczy (eyes)	0.060
about, with)	-	miał (had)	0.095	jakby (as if)	0.059
mu (him)	0.186	nad (over)	0.094	im (them)	0.059
dla (for)	0.185	żeby	0.094	mam (I have)	0.059
mi (me)	0.182	(in order to)	-	jestem (I am)	0.059
może (maybe)	0.180	ci (you)	0.092	oraz (and)	0.059
sobie (ourself)	0.179	powiedział (said)	0.091	ludzi (people)	0.058
ich (their)	0.178	potem (afterwards)	0.089	raz (one)	0.058
jeszcze (still)	0.169	u (at)	0.086	lat (years)	0.058

Table 4. Top of the 1-gram statistics of Polish collected from literature corpus of 949 371 453 words, (r.p. – reflexive pronoun, m. – masculin, f. – feminine, n. – neuter, pl. – plural, gen. – genitive). Approximated English translations are given in brackets

word (Eng.)	‰	word (Eng.)	%	word (Eng.)	%
chwili (moment)	0.575	głową (head)	0.423	proszę (please)	0.336
aż (till)	0.572	tę (this)	0.423	byli (were)	0.333
ona (she)	0.558	chwilę (moment)	0.411	czego (what)	0.331
wtedy (then)	0.548	dalej (farer)	0.411	pracy (woek)	0.330
no	0.547	ku (towards)	0.405	taki (such)	0.330
więcej (more)	0.543	mój (my)	0.402	ziemi (ground, earth)	0.329
mógł (could)	0.5437	zaś	0.390	czasie (time)	0.327
cię (you)	0.541	innych (others)	0.389	pierwszy (first)	0.327
między (between)	0.540	człowiek (human)	0.387	zaczął (started)	0.327
bardziej (more)	0.539	nikt (noone)	0.386	przykład (example)	0.326
nią (her)	0.531	dlatego (therefore)	0.385	wszystkim (all)	0.325
gdyby (if)	0.528	ktoś (someone)	0.384	człowieka (human)	0.325
roku (year)	0.527	powiedziała (said)	0.383	głos (voice)	0.325
których (which)	0.526	swoje (one's)	0.380	mogę (can, may)	0.324
również (also)	0.520	takie (such)	0.379	jakie (what)	0.324
czasu (time)	0.514	iż	0.378	musi (must)	0.323
wszystkie (all)	0.512	słowa (words)	0.378	temu (this)	0.323
jeden (one)	0.508	później (later)	0.377	prawie (almost)	0.323
wiem (know)	0.500	trochę (little)	0.375	trzy (three)	0.322
czym (what)	0.499	pana (master's)	0.368	znów (again)	0.321
wiele (many)	0.493	tyle (this much)	0.361	chciał (wanted)	0.319
którzy (which)	0.488	życie (life)	0.361	miejsce (place)	0.317
przecież (after all)	0.485	twarz (face)	0.360	myśli (think)	0.316
we (in)	0.481	szybko (fastly)	0.359	panie (sir)	0.314
czas (time)	0.481	końcu (end)	0.355	strony (pages,sides)	0.313
kto (who)	0.480	ponieważ (because)	0.351	obok (next to)	0.313
nam (us)	0.480	naprawdę (really)	0.351	zupełnie (absolutely)	0.313
wszyscy (all)	0.476	cały (whole)	0.350	powiedzieć (to say)	0.313
miała (had)	0.476	niech (let)	0.348	głowę (head)	0.313
kilka (a few)	0.475	jesteś (you are)	0.347	rzekł (said)	0.310
drzwi (doors)	0.473	dopiero (but,until)	0.347	mimo(despite)	0.310
wszystkich(all)	0.471	dzieci (childs)	0.344	nimi (them)	0.308
chyba (actually)	0.462	poza (apart from)	0.344	swego (own)	0.307
razem (together)	0.460	wreszcie (at last)	0.344	wielu(many)	0.306
którym (which)	0.453	którą (which)	0.342	ręce (hand)	0.305
dlaczego (why)	0.453	tutaj (here)	0.342	stronę (page, side)	0.303
której (which)	0.442	zbyt (too)	0.342	wciąż (still)	0.301
ludzie (people)	0.438	znowu (again)	0.341	coraz	0.301
nagle (suddenly)	0.438	oczywiście (of course)	0.341	moje (my)	0.297
dwa (two)	0.438	jeżeli (if)	0.341	dzień (day)	0.295
którego (which)	0.432	rzeczy (things)	0.340	pokoju (room, peace)	0.294
trzeba (need)	0.430	dnia (day)	0.340	mają (have)	0.294
choć (however)	0.429	jakiś (some)	0.337	każdy (each)	0.291
życia (life)	0.428	podczas (during)	0.337	prawda (true)	0.290
sobą (self)	0.428	ciebie (you)	0.336	został (became)	0.288

2-gram	‰	2-gram	‰	2-gram	‰
. nie	3.13	. jest	0.32	. dlaczego	0.20
. w	2.40	a potem	0.32	w którym	0.20
. a	1.69	. do	0.31	że jest	0.20
się w	1.51	mu się	0.31	. tylko	0.20
. to	1.49	w tej	0.31	. zapytał	0.20
. ale	1.24	. teraz	0.31	na pewno	0.20
. i	1.19	to co	0.30	na niego	0.20
się na	1.12	w końcu	0.29	i na	0.20
się z	1.05	do tego	0.29	po czym	0.20
. na	1.03	na przykład	0.29	jak to	0.20
się do	1.02	z tego	0.29	do domu	0.19
. tak	0.80	tak .	0.29	a nie	0.19
. z	0.74	z nich	0.29	. nic	0.19
. czy	0.71	po prostu	0.28	w ogóle	0.19
. po	0.71	. był	0.27	z tym	0.19
. co	0.68	to jest	0.27	co .	0.19
w tym	0.66	. za	0.27	nie mógł	0.19
się .	0.66	co się	0.26	nie był	0.19
się że	0.61	że w	0.26	. nawet	0.19
że nie	0.57	to że	0.26	się za	0.19
. jak	0.54	i tak	0.26	w ten	0.19
nie ma	0.53	i z	0.25	po raz	0.18
o tym	0.52	się o	0.25	nie może	0.18
. kiedy	0.50	się od	0.25	jak się	0.18
i nie	0.47	. potem	0.24	siebie .	0.18
się nie	0.46	. od	0.24	jeden z	0.18
się i	0.44	nic nie	0.24	. mam	0.18
nie jest	0.41	jest to	0.24	domu .	0.17
. o	0.41	nie tylko	0.24	. niech	0.17
to nie	0.41	. jego	0.24	jest w	0.17
. może	0.41	nie wiem	0.23	się to	0.17
na to	0.40	tak jak	0.23	. on	0.17
i w	0.40	. przez	0.23	w czasie	0.17
. no	0.40	w jego	0.23	do mnie	0.17
nie było	0.39	mnie .	0.22	. jestem	0.17
. jeśli	0.39	się po	0.22	nie będzie	0.17
ale nie	0.37	z nim	0.22	. oczywiście	0.17
nie .	0.37	i to	0.22	. w stronę	0.17
mi się	0.37	a w	0.21	a więc	0.17
że to	0.35	do niego	0.21	jak i	0.17
nigdy nie	0.34	głową .	0.21	po chwili	0.17
. gdy	0.34	. ten	0.21	. była	0.17
. ja	0.33	. już	0.21	w nim	0.17

Table 5. Top of the 2-gram statistics of Polish from a literature corpus

2-gram	‰	2-gram	‰	2-gram	‰
nikt nie	0.166	tym razem	0.142	z pewnością	0.124
. proszę	0.163	się tak	0.142	z nią	0.124
. spytał	0.162	na nią	0.142	wszystko co	0.123
. wszystko	0.162	od razu	0.142	. wiem	0.123
już nie	0.161	. więc	0.141	tym samym	0.122
. bo	0.161	i jego	0.141	z jego	0.120
na tym	0.160	tego co	0.141	powiedział .	0.120
ten sposób	0.160	poza tym	0.141	wcale nie	0.119
na mnie	0.160	ze sobą	0.140	. nagle	0.119
co to	0.159	go w	0.139	drzwi .	0.119
. jeżeli	0.158	to było	0.137	dlatego że	0.118
to .	0.157	. wszyscy	0.137	. dlatego	0.118
a ja	0.156	przez chwilę	0.137	a teraz	0.118
. nigdy	0.156	. jej	0.137	. miał	0.117
do siebie	0.155	aż do	0.137	się przez	0.117
nie miał	0.155	z powrotem	0.136	jak na	0.117
się jak	0.155	było to	0.136	co do	0.117
w stanie	0.154	dla mnie	0.135	w każdym	0.117
do niej	0.154	w ciągu	0.134	sobie że	0.117
na jego	0.153	. lecz	0.134	ale w	0.116
spojrzał na	0.153	nie można	0.134	. ty	0.116
za to	0.153	się nad	0.134	. nikt	0.116
. jeszcze	0.153	ale to	0.133	tak samo	0.115
wraz z	0.151	a może	0.133	ludzi .	0.115
może być	0.150	. pan	0.133	za nim	0.115
o to	0.150	ze mną	0.133	się stało	0.115
. gdyby	0.150	to w	0.133	nie była	0.115
czy nie	0.148	. jednak	0.132	niego .	0.114
to wszystko	0.148	w jej	0.132	jak w	0.114
. chyba	0.146	i że	0.132	. wtedy	0.114
czy to	0.146	. było	0.131	lat .	0.113
uśmiechnął się	0.146	a co	0.130	w kierunku	0.113
się ze	0.146	nie mam	0.129	w niej	0.112
przede wszystkim	0.145	. dobrze	0.129	podobnie jak	0.112
tym że	0.145	. kto	0.128	w sobie	0.112
jest .	0.145	. dla	0.127	odezwał się	0.112
nie mogę	0.145	jeszcze nie	0.127	już w	0.111
w domu	0.144	dobrze .	0.126	go do	0.111
. przecież	0.144	. ze	0.126	z tych	0.111
być może	0.144	. ta	0.125	w której	0.111
oczy .	0.143	. bardzo	0.125	życia .	0.111
prawda .	0.143	się już	0.125	nadzieję że	0.110
tego nie	0.143	a także	0.124	dalej .	0.110
był to	0.142	to znaczy	0.124	. gdzie	0.110

2-gram	‰	2-gram	‱	2-gram	‱
. mimo	0.109	się pod	0.970	wszystko .	0.889
tej chwili	0.109	w takim	0.969	. przed	0.888
. przy	0.109	że się	0.969	. zawsze	0.887
nawet nie	0.109	do tej	0.966	. mój	0.886
z powodu	0.109	w porządku	0.966	to samo	0.884
a to	0.108	. jesteś	0.962	nim .	0.880
go .	0.108	. tym	0.961	. powiedział	0.879
. cóż	0.107	. coś	0.956	ale i	0.878
to się	0.107	o czym	0.952	. te	0.877
. tu	0.106	o czym	0.952	od czasu	0.875
można było	0.106	na chwilę	0.950	ziemi .	0.872
w życiu	0.106	. sam	0.950	jak gdyby	0.870
z nimi	0.106	. tam	0.945	ci się	0.870
raz pierwszy	0.105	chodzi o	0.939	. dopiero	0.867
i co	0.105	to był	0.939	podczas gdy	0.867
a na	0.104	. są	0.939	. muszę	0.865
odwrócił się	0.104	. pod	0.938	. jeden	0.864
tym co	0.103	. poza	0.935	było .	0.864
. trzeba	0.103	nie są	0.933	w pobliżu	0.862
i po	0.103	razem z	0.932	. bez	0.861
w dół	0.103	na ziemi	0.932	i do	0.860
wiem .	0.103	o co	0.929	życie .	0.858
się jej	0.102	zgodnie z	0.929	zrobić .	0.858
od tego	0.102	z tobą	0.928	jeszcze raz	0.856
na temat	0.102	się jeszcze	0.927	na siebie	0.853
a nawet	0.101	za sobą	0.923	więcej niż	0.852
ja .	0.101	był w	0.920	w których	0.847
po co	0.101	to na	0.919	. spytała	0.843
do nich	0.101	do końca	0.918	. wreszcie	0.841
w górę	0.101	sobie sprawę	0.918	tylko w	0.841
. właśnie	0.100	to tylko	0.917	co z	0.841
względu na	0.100	myślę że	0.914	to z	0.838
się przed	0.100	by się	0.913	stało się	0.838
była to	0.100	. ona	0.908	. tego	0.836
go na	0.100	nie mogła	0.908	się tylko	0.834
wiedział że	0.100	i o	0.905	że na	0.833
jednym z	0.099	. pani	0.903	. albo	0.830
przy tym	0.099	jednak nie	0.901	po to	0.829
go nie	0.099	tak się	0.900	i jak	0.827
. och	0.098	dla niego	0.900	między innymi	0.823
na jej	0.098	tak że	0.893	mimo to	0.823
nie jestem	0.098	czy też	0.893	i ja	0.822
jeśli nie	0.097	stało .	0.891	w polsce	0.819
to już	0.097	. ponieważ	0.890	w swoim	0.815

3-gram	‱	3-gram	‱	3-gram	‱
w ten sposób	1.71	się z nim	0.613	. dlaczego .	0.480
. tak .	1.59	. no i	0.608	do siebie .	0.480
. nie .	1.55	. nie mogę	0.607	w tym momencie	0.479
. nie wiem	1.32	. nie mam	0.598	. nic nie	0.477
. w tym	1.30	od czasu do	0.593	to nie jest	0.475
. nie ma	1.23	czasu do czasu	0.592	że nie ma	0.474
po raz pierwszy	1.13	w tym samym	0.592	po drugiej stronie	0.473
. to nie	1.10	. tym razem	0.589	. to co	0.472
. w końcu	1.07	o tym że	0.585	w tym czasie	0.472
. ale nie	1.05	sobie sprawę że	0.582	w ogóle nie	0.470
. a więc	0.998	. w każdym	0.573	. a jeśli	0.469
w tej chwili	0.985	. na pewno	0.572	. w takim	0.468
. a co	0.945	. i to	0.572	. przez chwilę	0.468
. czy to	0.911	. a ja	0.565	. po co	0.466
na to że	0.901	. i nie	0.559	. co .	0.464
. a może	0.821	w takim razie	0.552	. i co	0.461
w każdym razie	0.813	. nie nie	0.5525	. nie jestem	0.460
. po chwili	0.811	się do niego	0.550	. a ty	0.457
. poza tym	0.807	w jaki sposób	0.546	nie ma .	0.457
. nigdy nie	0.782	. nikt nie	0.539	do tej pory	0.446
. nie było	0.775	wydaje mi się	0.529	. wiem że	0.446
mi się że	0.764	w porządku .	0.528	. jak się	0.445
. a teraz	0.762	. na przykład	0.524	. a jednak	0.442
. był to	0.757	w stosunku do	0.516	. to był	0.442
do domu .	0.751	mam nadzieję że	0.513	. mam nadzieję	0.441
. być może	0.739	. w ten	0.510	. niech pan	0.437
. w tej	0.733	. tak więc	0.507	o tym .	0.435
. jest to	0.725	. to znaczy	0.507	. mimo to	0.434
ze względu na	0.710	. no to	0.506	. nie chcę	0.431
co się stało	0.707	tak samo jak	0.506	co się dzieje	0.429
. co to	0.704	. a to	0.506	. no cóż	0.428
. a potem	0.703	. była to	0.505	do wniosku że	0.416
. po prostu	0.688	okazało się że	0.504	się z nią	0.408
. myślę że	0.676	. jeden z	0.501	. nie jest	0.403
. ale to	0.660	. było to	0.497	się do niej	0.399
. co się	0.657	w związku z	0.497	o tym nie	0.399
. i tak	0.657	z drugiej strony	0.496	za każdym razem	0.398
się stało .	0.653	zwrócił się do	0.495	. spojrzał na	0.394
nie wiem .	0.641	nie było .	0.489	na pewno nie	0.393
. to jest	0.639	. czy nie	0.488	. uśmiechnął się	0.385
. jak to	0.626	. to było	0.486	. po raz	0.320

Table 6. Top of the 3-gram statistics of Polish from a literature corpus. They are very good data to model language but are difficult to be collected for inflected languages in amount which is enough for applications. The model we manage to build seems to be large enough to properly describe language by statistics of 3-grams

3-gram	‱	3-gram	‱	3-gram	‱
z tego co	0.319	. wszystko to	0.282	się na to	0.252
jeśli chodzi o	0.319	to wszystko .	0.282	i tak dalej	0.251
. wiedział że	0.319	. i w	0.282	. ale co	0.251
. może to	0.317	się w nim	0.281	się o tym	0.250
po to by	0.317	. o co	0.281	się o tym	0.250
na ziemię .	0.317	. cit .	0.280	w każdej chwili	0.250
. tak to	0.316	. ale ja	0.279	. a przecież	0.249
. to wszystko	0.316	to prawda .	0.278	. nie tylko	0.248
odwrócił się i	0.316	jak to się	0.276	. okazało się	0.247
. to prawda	0.315	w gruncie rzeczy	0.276	względu na to	0.247
się ze mną	0.315	. podobnie jak	0.275	się w jego	0.247
. dobrze .	0.314	. ja nie	0.274	udało mi się	0.245
. odwrócił się	0.312	mu się że	0.274	pokręcił głową .	0.244
udało mu się	0.311	w porównaniu z	0.274	. po pierwsze	0.241
że jest to	0.310	. z drugiej	0.271	. no tak	0.241
. oczywiście .	0.309	na ziemi .	0.271	w dalszym ciągu	0.241
za to że	0.308	z tego powodu	0.270	. a zatem	0.241
przed naszą erą	0.308	w chwili gdy	0.269	. nie to	0.241
nie da się	0.308	w dół .	0.269	. spojrzała na	0.241
. nie miał	0.307	się dzieje .	0.268	. od czasu	0.240
. ja .	0.307	na przykład w	0.267	na zewnątrz .	0.240
to znaczy że	0.306	. w jego	0.267	się z tobą	0.239
. jeśli nie	0.304	. zdaje się	0.266	po co .	0.239
. dlaczego nie	0.303	. oczywiście że	0.259	z dala od	0.238
się że to	0.301	. przede wszystkim	0.259	sobie sprawę z	0.238
w tym miejscu	0.301	obawiam się że	0.259	. nawet nie	0.238
do czynienia z	0.301	. uśmiechnęła się	0.258	. przykro mi	0.237
. to była	0.300	. chyba nie	0.258	. to właśnie	0.237
się w stronę	0.299	z nich .	0.258	na to .	0.237
po prostu nie	0.298	że nie jest	0.258	się do nich	0.236
z tego że	0.297	o tym jak	0.258	. nie rozumiem	0.236
na mnie .	0.297	nie można było	0.258	wygląda na to	0.236
nie wiem czy	0.295	z nich nie	0.257	co chodzi .	0.235
co to za	0.294	w taki sposób	0.257	. zgodnie z	0.234
się w tym	0.293	wpatrywał się w	0.256	. naprawdę .	0.234
to jest .	0.293	się nad tym	0.255	jest w stanie	0.233
na niego .	0.292	do drzwi .	0.255	. co ty	0.232
że to nie	0.290	. jeszcze nie	0.255	. i wtedy	0.232
. o ile	0.290	tylko dlatego że	0.254	do niej .	0.231
. nie był	0.289	i spojrzał na	0.254	tej samej chwili	0.231
. sądzę że	0.289	z powrotem .	0.253	. wydaje mi	0.230
za nim .	0.289	w tej sprawie	0.253	i z powrotem	0.230
. nie można	0.285	w przeciwieństwie do	0.253	. do tego	0.229
. a kiedy	0.284	się z nimi	0.253	odwrócił się do	0.228
z jednej strony	0.284	ale to nie	0.252	. nie sądzę	0.228
nie wiem co	0.282	nie mógł się	0.252	to samo .	0.228

3-gram	‱	3-gram	‱	3-gram	‱
z pewnością nie	0.228	a co z	0.210	. wygląda na	0.195
. spytał .	0.228	spojrzał na nią	0.210	w jednym z	0.195
potrząsnął głową .	0.228	. dlatego też	0.209	w odniesieniu do	0.194
nie ma w	0.227	. co prawda	0.209	w jakiś sposób	0.193
się coraz bardziej	0.227	nigdy się nie	0.208	w zależności od	0.192
wydaje się że	0.226	. a czy	0.208	. nie mogła	0.192
po tym jak	0.226	przez jakiś czas	0.207	po raz ostatni	0.192
czy nie .	0.225	. o czym	0.207	związku z tym	0.192
nie było to	0.225	o to że	0.207	. co z	0.191
. tak się	0.224	. wcale nie	0.207	zdawało się że	0.191
z tobą .	0.224	. no .	0.206	. nie wolno	0.191
w górę .	0.224	na myśli .	0.206	. wiem .	0.190
wydawało mi się	0.224	na zawsze .	0.206	. po czym	0.190
za późno .	0.224	. no więc	0.206	do przodu .	0.190
nie jest w	0.224	na to co	0.206	po raz drugi	0.190
. obawiam się	0.222	nie żyje .	0.205	do pracy .	0.190
co to znaczy	0.222	. co do	0.205	. na tym	0.189
nie ma nic	0.222	. to ja	0.204	. o tym	0.189
po obu stronach	0.221	w miarę jak	0.204	tylko po to	0.189
nie było w	0.221	. ale jak	0.203	. ale .	0.189
nie wiem jak	0.220	. tak czy	0.203	to zrobić .	0.188
. wydaje się	0.219	nic z tego	0.202	. to się	0.188
. to już	0.219	się w niej	0.202	że nigdy nie	0.188
był w stanie	0.218	. ale teraz	0.202	do tego że	0.187
. za to	0.218	po to żeby	0.202	się nie stało	0.186
. myślałem że	0.218	. to tylko	0.202	. zdziwił się	0.186
jak na przykład	0.217	. po kilku	0.202	. nie była	0.186
znalazł się w	0.217	. przecież to	0.200	na miejscu .	0.185
. na to	0.217	. to bardzo	0.200	. nie możemy	0.185
coś w rodzaju	0.217	i w tym	0.200	o tej porze	0.185
ze sobą .	0.216	raz po raz	0.199	. przez cały	0.185
na świecie .	0.216	z punktu widzenia	0.199	wyglądało na to	0.185
co się z	0.215	zbliżył się do	0.199	. zastanawiał się	0.185
spojrzała na niego	0.215	znajduje się w	0.199	na drugą stronę	0.184
. gdyby nie	0.215	to co się	0.199	w ostatniej chwili	0.184
. no dobrze	0.214	z powrotem na	0.199	. a poza	0.184
z nim .	0.213	do głowy .	0.199	w milczeniu .	0.184
. wydawało się	0.213	w tym celu	0.199	. tak samo	0.184
z całą pewnością	0.213	. co więcej	0.199	i w ogóle	0.184
i tak nie	0.213	. kiedy się	0.198	. nie mogłem	0.183
wszystko w porządku	0.212	nie mam pojęcia	0.198	. nie będę	0.183
z tego .	0.212	się z tego	0.197	co z tego	0.183
na niego z	0.212	się że w	0.197	co do tego	0.183
nie był w	0.211	. ale czy	0.196	. chodzi o	0.183
na to nie	0.211	w głowie .	0.195	. ale przecież	0.182
. myślisz że	0.210	na wszelki wypadek	0.195	niezależnie od tego	0.182

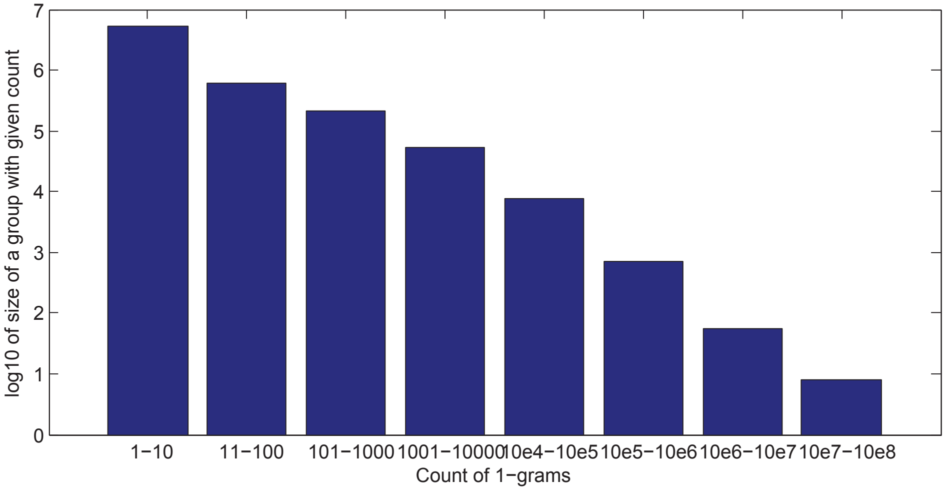


Fig. 2. Histogram of 1-grams (in logarithm scale). There are many 1-grams which are very rare. The amount goes down with increasing count of a 1-gram. The histograms of 2- and 3-grams are very similar

5. Implementation and applications

Storing large vocabulary n-gram model is another issue to concern. 2- and 3-grams cannot be stored as strings because they would use too much disk space. This is why each 1-gram (unigram on Fig. 3) has an ID. The 2-grams are stored as two 1-gram ID, which are integer numbers. The each 2-gram has its id_bigram, so 3-grams are stored as a set of two id_bigrams. The language properties have been very often modelled by n-grams (Huang & Lippman, 1988), (Young et al., 2005), (Manning, 1999), (Jurafsky & Martin, 2008), (Khudanpur & Wu, 1999), (Whittaker & Woodland, 2003), (Hirsimaki et al., 2009). Let us assume the word string $w \in W$ consisting of n words $w_1, w_2, w_3, ..., w_n$. Let $P(W)$ be a set of probability distributions over possible word strings W that reflects how often $w \in W$ occurs. It can be decomposed as

$$P(w) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)...P(w_n|w_1, ..w_{n-1}). \tag{1}$$

It is theoretically justified and practically useful assumption that, $P(w)$ dependence is limited to n words backwards. Probably the most popular are trigram models where $P(w_i|w_{i-2}, w_{i-1})$, as a dependence on the previous two words is the most important, while model complication is not very high. Such models still need statistics collected over a vast amount of text. As a result many dependencies can be averaged. Simplified case of applying n-grams in speech recognition is presented in Fig. 4.

N-grams are the most basic and common language model in ASR systems (Young et al., 2005), (Lamere et al., 2004), (Whittaker & Woodland, 2003), (Hirsimaki et al., 2009). It is a result of their simplicity end effectiveness Our attempt was to build such model for large vocabulary Polish applications. The large number of analysed texts will allow us to predict words being recognised and improve recognition of the ASR system highly.

Polish is highly inflected in comparison to English. The rich morphology causes difficulties in training language models due to data sparsity. Much more text data must be used for inflected

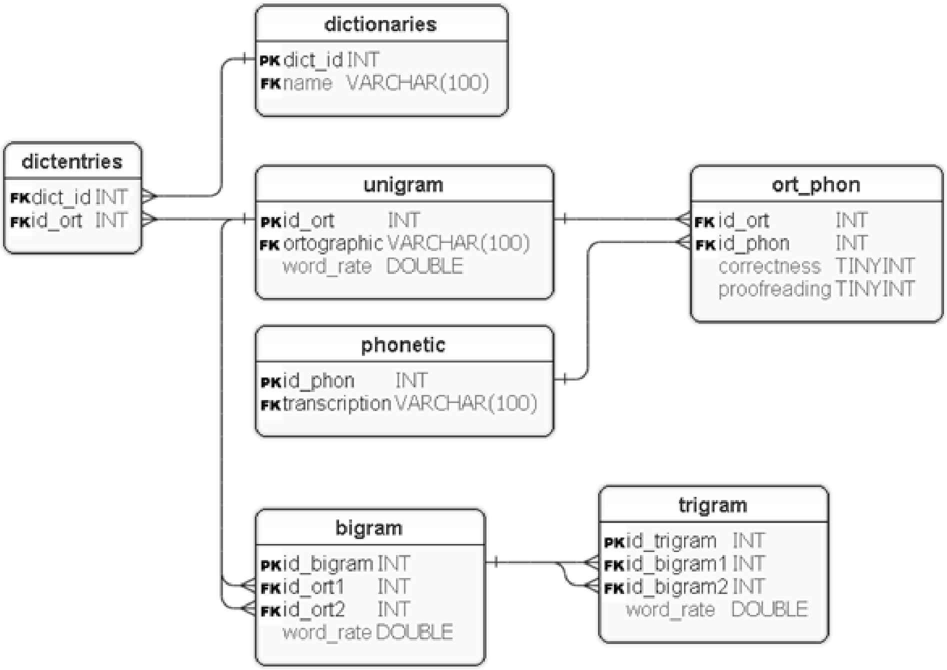


Fig. 3. Our n-gram model is a part of a dictionary implemented in SQL.

languages than for positional ones to achieve the model of the same efficiency (Whittaker & Woodland, 2003).

The modified weighted Levenshtein distance (MWLD) (Ziółko, Gałka, Skurzok & Jadczyk, 2010) and dynamic time warping (DTW) (Rabiner & Juang, 1993) algorithms allow to evaluate a distance of words from an ASR system dictionary with a sequence of phoneme hypotheses. In case of recognising continuous speech, this procedure have to be repeated hundreds thousands of time for different words and different phoneme hypotheses. An optimal decision is taken to find a sequence of word hypotheses. This processes is known as level builder.

Typically, the situation is even more complex. Instead of a sequence of words, a lattice of words should be built. The final sentence hypothesis is taken from the lattice, by applying syntax and semantic modelling.

Word hypotheses are sorted by natural logarithms of MWLD or DTW. The W words with lowest distances are introduced to the lattice for each allowed start point of a word.

Let us assume a set of I word hypotheses and matrix $H \in (C, \Re)^{n \times k}$ of phoneme hypotheses where C stands for a set of characters representing Polish phonemes, $\Re$ are logaritms of propabilities, n is size of C (number of possible phoneme types) and k corresponds to time. Let us introduce w_m as m-th word of a M size $(0 < m \leq M)$ dictionary. Then, let us denote a_i as a start time of ith word hypothesis and b_i as its end. Let us introduce $p_m(a_i = t_1, b_i = t_2)$ as a probability that word w_m is an ith observation for a sequence of phonemes from time a_i to

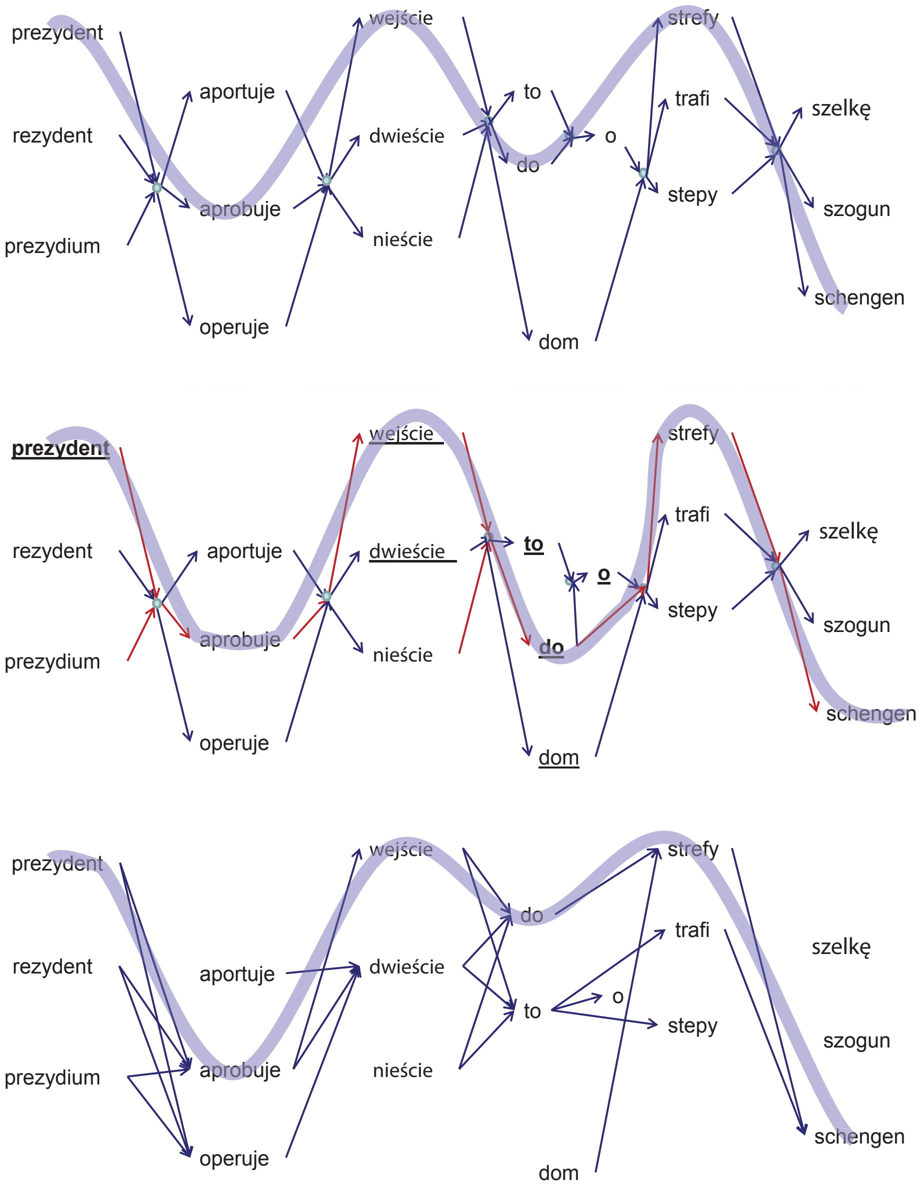


Fig. 4. The general word lattice is presented in the upper diagram. A lattice with stressing of probable 1 grams (bold and undelined) and 2 grams (red arrows) is depicted in the middle one. A word lattice with reduction of unprobable 2-grams is shown in the bottom one. In all cases the correct sentence is marked by a purple shadow. In the second case it leads mainly via strong n-grams. In the third case the proper path still exists in the lattice after reductions

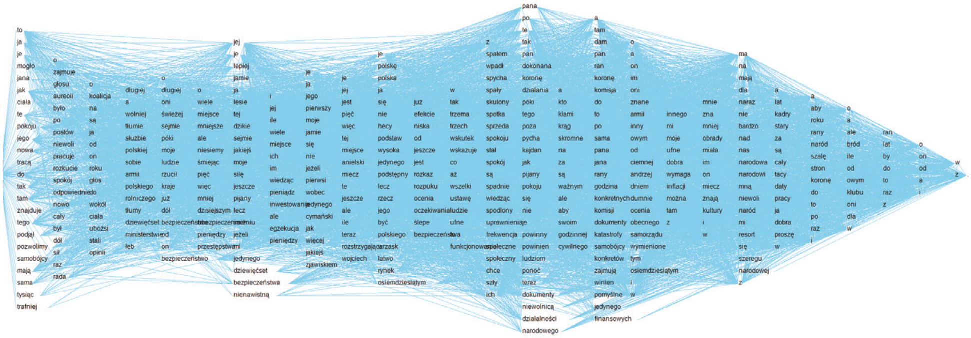


Fig. 5. Real word lattice generated by AGH ASR system shows complexity of the graph and importance of applying language modelling like n-grams

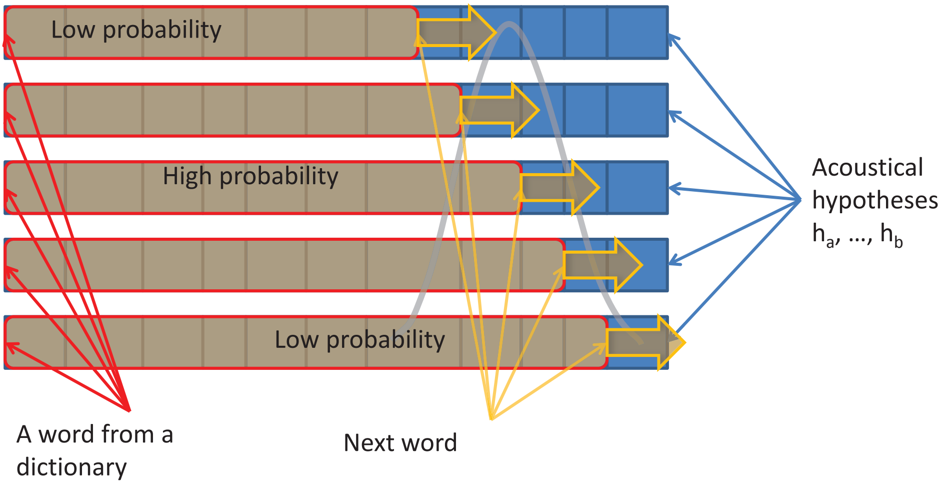


Fig. 6. A level builder fits a dictionary word into acoustical hypotheses on different time scales

time b_i such as

$$a_i = \begin{cases} 1 & \text{for the words following a starting node} \\ b_{i-j} + 1 \pm t & \text{for others} \end{cases}, \tag{2}$$

where b_{i-j} is an end of another word hypothesis and where $t = 3$ is a threshold of allowed time distance between neighbouring words counted in the number of frames (phoneme hypotheses). In the simplest case $j = 1$, but generally $j < i$ (in case of a lattice). The task of level building is to maximise $p_m(a_i = t_1, b_i = t_2)$ by changing m, a_i and b_i. Difference $b_i - a_i$ is constant for a particular word w_m and there are restrictions for a_i described above (a_i of a word has to follow b_{i-j} of another word in time domain). Typically there are between 10

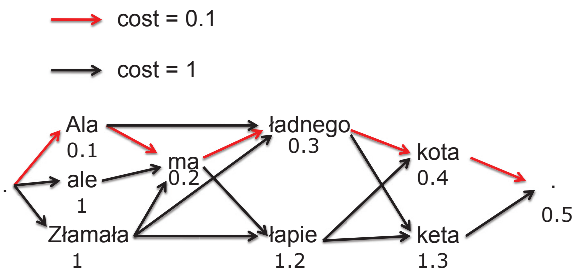
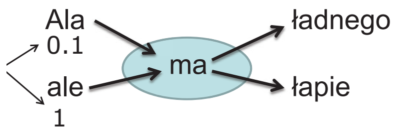


Fig. 7. Simple example of word network showing usage of 2-grams to find the best path. The words in the lattice mean: Ala – female name, ale – but, Złamała – broke (feminine), ma – has, ładnego – pretty (masculin), łapie – catches, kota – a cat (accusativus), keta – a chain (in silesian dialect)



next	cost
ładnego	0.2
łapie	1.1

Fig. 8. Example of calculating weights for a word using 3-grams. Its possible weights are based on words preceding and following in the word network. English translations are in Fig. 7

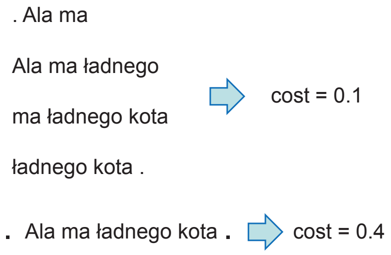
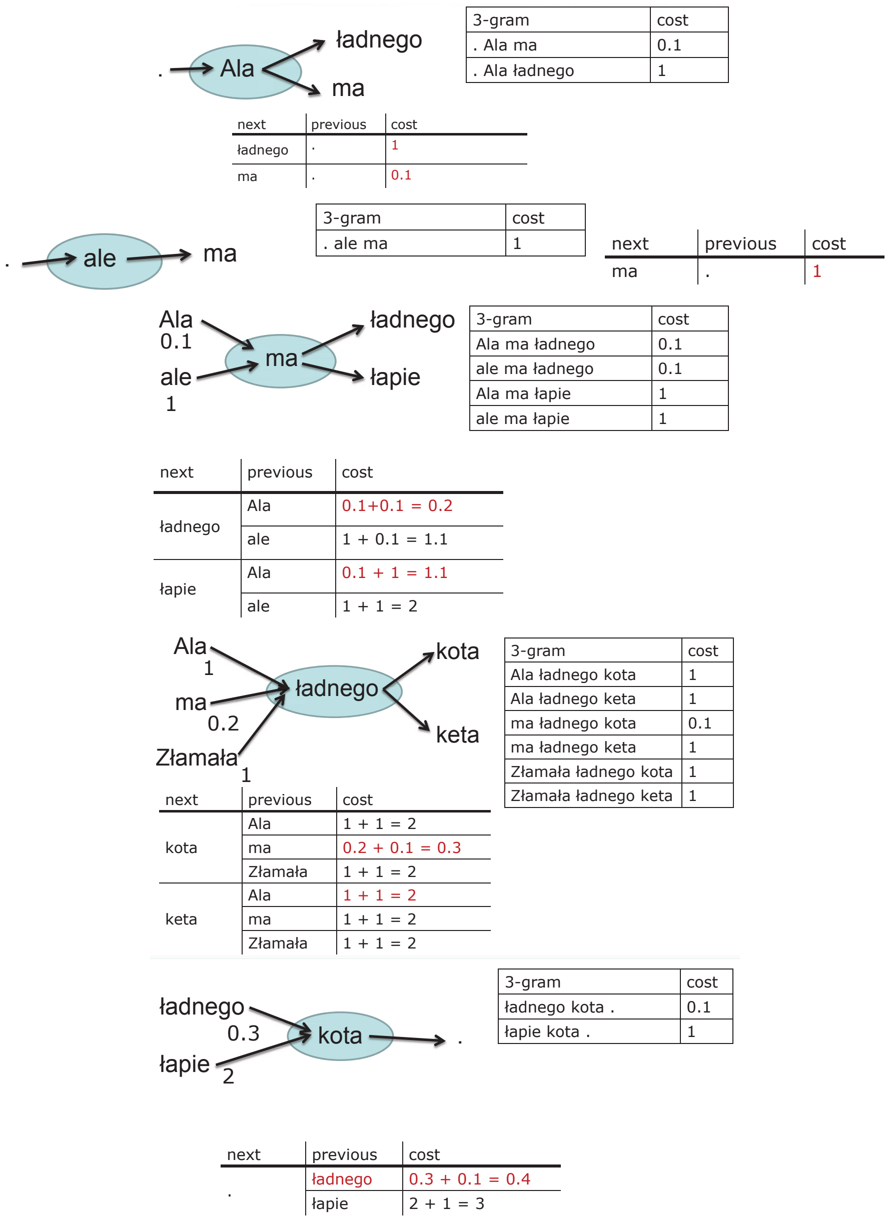


Fig. 9. 3-grams used to decode a sentence from the example from Fig. 8. English translations are in Fig. 7



3-gram	cost
. Ala ma	0.1
. Ala ładnego	1

next	previous	cost
ładnego	.	1
ma	.	0.1

3-gram	cost
. ale ma	1

next	previous	cost
ma	.	1

3-gram	cost
Ala ma ładnego	0.1
ale ma ładnego	0.1
Ala ma łapie	1
ale ma łapie	1

next	previous	cost
ładnego	Ala	0.1+0.1 = 0.2
	ale	1 + 0.1 = 1.1
łapie	Ala	0.1 + 1 = 1.1
	ale	1 + 1 = 2

3-gram	cost
Ala ładnego kota	1
Ala ładnego keta	1
ma ładnego kota	0.1
ma ładnego keta	1
Złamała ładnego kota	1
Złamała ładnego keta	1

next	previous	cost
kota	Ala	1 + 1 = 2
	ma	0.2 + 0.1 = 0.3
	Złamała	1 + 1 = 2
keta	Ala	1 + 1 = 2
	ma	1 + 1 = 2
	Złamała	1 + 1 = 2

3-gram	cost
ładnego kota .	0.1
łapie kota .	1

next	previous	cost
.	ładnego	0.3 + 0.1 = 0.4
	łapie	2 + 1 = 3

Fig. 10. Proces of finding the best path through the word network using 3-gram weights, node after a node. English translations are in Fig. 7

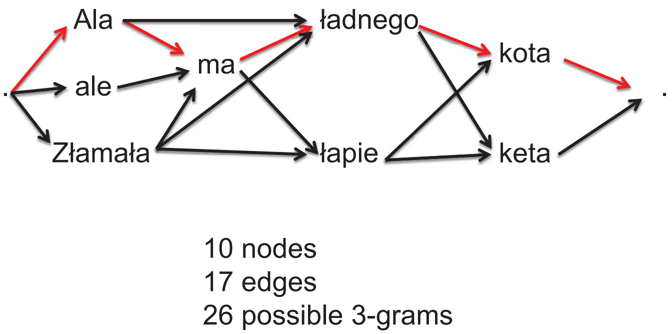


Fig. 11. Result of searching for the best path through the word network using 3-gram weight, node after a node (see Fig. 10). English translations are in Fig. 7

and 50 parallel word hypotheses allowed to start from a particular time point in the described way.

The word hypotheses are turned into a lattice by connecting nodes if ends and starts are closer to each other in time then a chosen threshold.

Created word lattices are large, which makes searching for a best path time consuming, while ASR system should work in real time. This is why, edges which statistically were found unlikely by n-grams can be cut out.

Finding the best path can be provided using Dijkstra algorithm (Dijkstra, 1959). Applying 2-grams is very straightforward, but using 3-grams is more complex. This is why we will disscuss its possible implementation considering an example. The whole network of our example is presented in Fig. 7, but with simplified values from 2-grams only. Then calculating probability for a particular word using 3-grams is presented in Fig. 8. It has to be stressed that many more calculations have to be conducted to calculate these weights, and also many more values have to be kept when the best path is searched. Fig. 9 shows the entire sentence we want to decode and its weights using 3-grams being components of this sentence. Fig. 10 shows searching the best path node after a node. Our example has 10 nodes and 17 edges. It results in 26 possible 3-grams (Fig. 11).

Typically n-grams of higher orders are smoothed by backing-off methods (Kneser & Ney, 1995; Ney et al., 1994). It can improve results by up to 5%. Another recently popular method is to apply Bloom filter (Bloom, 1970) instead of backing-off.

The presented n-gram model of Polish will be licensed to be available for both research and commercial applications. The first commercial usage will be an Imaginary Interface made by SnapKeys. It has 4 imaginary letter keys at the begining. Afterwards a user can hide them because they can begin to blind type anywhere on the screen. It leaves entire scrren for displaying output data and allow faster typing thanks to smaller finger movements. The interface connects several probability models to find words which a user wants – 1-gram being one of them. The Polish version is now being developed using our model.

6. Conclusions

N-gram models are strightforward but very effective in language modelling. Large corpora are necessary to build effective n-grams models. This and other problems make this task especially complicated for languages like Polish which are highly inflected and without very large professional text corpora. Eventhough this difficulties, a succesful n-grams model of Polish was build at AGH and offered to public.

7. Acknowledgments

This work was supported by MNISW grant number OR00001905.

8. References

Bloom, B. H. (1970). Space/time trade-offs in hash coding with allowable errors, *Communications of the ACM* **13, 7**.

Dijkstra, E. (1959). A note on two problems in connexion with graphs, *Numerische Mathematik* 1: 269–271.

Hirsimaki, T., Pylkkonen, J. & Kurimo, M. (2009). Importance of high-order n-gram models in morph-based speech recognition, *IEEE Transactions on Audio, Speech and Language Processing* 17(4): 724–32.

Huang, W. & Lippman, R. (1988). Neural net and traditional classifiers, *Neural Information Processing Systems, D. Anderson, ed.* pp. 387–396.

Jurafsky, D. & Martin, J. H. (2008). *Speech and Language Processing, 2nd Edition*, Prentice-Hall, Inc., New Jersey.

Khudanpur, S. & Wu, J. (1999). A maximum entropy language model integrating n-grams and topic dependencies for conversational speech recognition, *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Phoenix, AZ* .

Kneser, R. & Ney, H. (1995). Improved backing-off for m-gram language modelling, *Proceedings of International Conference on Acoustics, Speech, and Signal Processing, ICASSP* pp. 181–184.

Lamere, P., Kwok, P., Gouvea, E., Raj, B., Singh, R., Walker, W. & Wolf, P. (2004). The CMU Sphinx-4 speech recognition system, *Sun Microsystems* .

Manning, C. D. (1999). *Foundations of Statistical Natural Language Processing*, MIT Press. Cambridge, MA.

Ney, H., Essen, U. & Kneser, R. (1994). On structuring probabilistic dependencies in stochastic language modelling, *Computer Speech and Language* 8: 1–38.

Przepiórkowski, A. (2004). *The IPI PAN Corpus: Preliminary version*, IPI PAN, Warszawa.

Rabiner, L. & Juang, B. H. (1993). *Fundamentals of speech recognition*, PTR Prentice-Hall, Inc., New Jersey.

Whittaker, E. & Woodland, P. (2003). Language modelling for Russian and English using words and classes, *Computer Speech and Language* 17: 87–104.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V. & Woodland, P. (2005). *HTK Book*, Cambridge University Engineering Department, UK.

Ziółko, B., Gałka, J., Skurzok, D. & Jadczyk, T. (2010). Modified weighted Levenshtein distance in automatic speech recognition, *Proceedings of XVI KKZMBM* pp. 116–120.

Ziółko, B., Skurzok, D. & Michalska, M. (2010). Polish n-grams and their correction process, *Proceedings of The 4th International Conference on Multimedia and Ubiquitous Engineering (MUE 2010), Cebu, Philipines* .

Ziółko, B., Skurzok, D. & Ziółko, M. (2010). Word n-grams for polish, *The Tenth IASTED International Conference on Artificial Intelligence and Applications, AIA 2010* .

Part 4

Text to Speech Systems and Emotional Speech

Multilingual and Multimodal Corpus-Based Text-to-Speech System – PLATTOS –

Matej Rojc[1] and Izidor Mlakar[2]
[1]*Faculty of Electrical Engineering and Computer Science, University of Maribor,*
[2]*Roboti C.S.,*
Slovenia

1. Introduction

Over the last decade a lot of TTS systems have been developed around the world that are more or less language-dependent and more or less time and space-efficient (Campbell & Black, 1996; Holzapfel, 2000; Raitio et al., 2011; Sproat, 1998; Taylor et al., 1998). However, speech technology-based applications demand time and space-efficient multilingual, polyglot, and multimodal TTS systems. Due to these facts and due to the need for a powerful, flexible, reliable and easily maintainable multimodal text-to-speech synthesis system, a design pattern is presented that serves as a flexible and language independent framework for efficient pipelining all text-to-speech processing steps. The presented design pattern is based on time and space-efficient architecture, where finite-state machines (FSM) and heterogeneous relation graphs (HRG) are integrated into a common TTS engine through the so-called ''queuing mechanism''. FSMs are a time-and-space efficient representation of language resources and are used for the separation of language-dependent parts from the language-independent TTS engine. On the other hand, the HRG structure is used for storing all linguistic and acoustic knowledge about the input sentence, for the representation of very heterogeneous data and for the flexible feature constructions needed by various machine-learned models that are used in general TTS systems. In this way, all the algorithms in the presented TTS system use the same data structure for gathering linguistic information about input text, all input and output formats between modules are compatible, the structure is modular and interchangeable, easily maintainable and object oriented (Rojc & Kačič, 2007). The general idea of corpus-based speech synthesis is the use of a large speech corpus for acoustic inventory and for creating realistic-sounding, machine-generated speech from raw waveform segments that are directly concatenated without any or only minimal signal processing. Since only a limited size speech corpus can be used, a compromise between the number of speech units in different prosodic contexts and the overall corpus size should normally be reached. On the other hand, the unit selection algorithm has to select the most suitable sequence of units from the acoustic inventory, where longer units should be favoured. Namely, when using longer units, the number of concatenation points can be reduced, resulting in more natural synthetic speech. The performance of the overall unit selection algorithm for corpus-based synthesis, regarding quality and speed, depends on the solving of several issues, e.g. preparation of text corpus, acoustic inventory construction using non-uniform units, reduction of unit search space, detection and removal of acoustically very similar units, off-line calculation of concatenation costs between all speech units in the acoustic inventory, their efficient representation, and their fast access within the on-line system. Further, the optimisation of weights used within cost function is an important issue, since these weights mainly influence the unit selection process performance regarding synthesised speech quality and naturalness (Black et al., 1997; Christophe et al., 2002). In the presented design pattern for corpus-based TTS systems, a gradient descent based unit selection optimisation algorithm is proposed for optimising unit cost functions' weights. Furthermore, the presented unit selection process addresses issues, such as: efficient acoustic inventory construction, reduction of unit search space, detection and removal of acoustically similar units, calculation of the concatenation costs, efficient representation of concatenation costs, and fast lookup. An important aspect of the presented cost functions' weights optimisation is that it also reduces laborious manual involvement when preparing new voices and tuning the best possible quality of the corpus-based TTS system. No matter what age, cultural background, or even what language people might speak, facial expressions and different body gestures always occur in natural human-human dialogues. Even when the dialogue is not face-to-face, people are prone to describing key issues by using different facial expressions or even by hands that remain free. Therefore, the first reason for using non-verbal modalities together with the TTS system, is to better emulate the natural course of the dialogue, and to make people feel more comfortable when ''speaking'' to a machine. The second reason is hidden in those issues that occur during the usage of human-machine interaction systems. The need to repeat and the misinterpretation of speaking terms are common features regarding the majority of users. Such behaviour usually leads towards less-functional and less-efficient spoken dialogue systems (Cassell, 2000). If we were to have more appropriate social responses from the machine through personification of the TTS system by using embodied conversational agents (ECA), people will more readily respond with emotive socially-coloured responses. Therefore, human-human-like communicative behaviour may be evoked in this way, giving the spoken dialogue system the ability to shape and adjust the dialogue to its own rules. TTS systems and believable characters (ECAs) can be used together to evoke communicative behaviour. ECAs can often, by expressing social tendencies, shape and also lead the dialogue. Understanding of attitude, emotion, together with how gestures (facial and hand) and body movements complement, or in some cases, override any verbal information produced by the TTS system thus providing crucial information for modelling both the dialogue and the ECA's socially-oriented responses. The social response (naturality) of the TTS system fused with ECA can then be presented to the user in a more human-like form, using not just audio but also facial expressions, such as: facial emotions, visual animation of synthesised speech, and correlated head, hand, and body movements. Therefore, personificated TTS systems enable the development of more advanced, personalized, and more natural multimodal-output-based human-machine interfaces that are in demand more and more for today's applications and environments. The time and space-efficient architecture of the corpus-based TTS system is presented in Section 2. The unit selection process for corpus-based TTS systems is then described in Section 3. The next section describes in detail the novel EVA framework that enables personification of the general TTS system. Slovenian implementation of the multilingual and multimodal corpus-based PLATTOS TTS system is presented in Section 5. A novel approach to distributive evaluation and the testing of TTS systems is presented in Section 6. Conclusions are drawn at the end.

2. Time and space-efficient TTS architecture

The corpus-based TTS architecture of the PLATTOS TTS system presented in Figure 1 is modular, time and space-efficient, and flexible (Rojc, 2003; Rojc & Kačič, 2007). By following the multilingual aspect, the language-dependent resources are separated from the language-independent core TTS engine. Its modular structure allows for all modules within the system to be easily maintained, and further improved by easy integration of new algorithms into the TTS system.

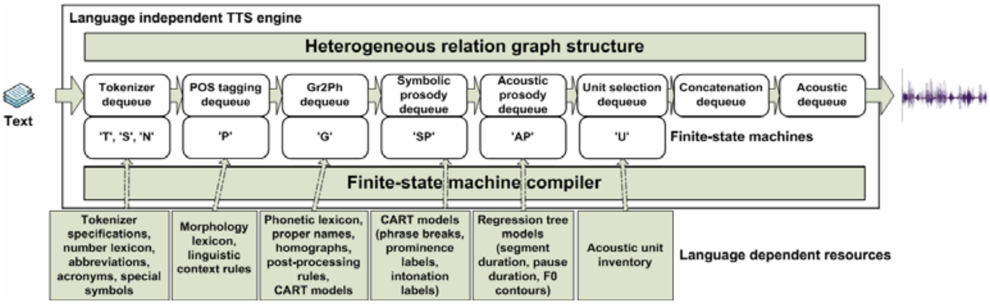

Language independent TTS engine
Heterogeneous relation graph structure
Text
Tokenizer dequeue
POS tagging dequeue
Gr2Ph dequeue
Symbolic prosody dequeue
Acoustic prosody dequeue
Unit selection dequeue
Concatenation dequeue
Acoustic dequeue
'T', 'S', 'N'
'P'
'G'
'SP'
'AP'
'U'
Finite-state machines
Finite-state machine compiler
Tokenizer specifications, number lexicon, abbreviations, acronyms, special symbols
Morphology lexicon, linguistic context rules
Phonetic lexicon, proper names, homographs, post-processing rules, CART models
CART models (phrase breaks, prominence labels, intonation labels)
Regression tree models (segment duration, pause duration, F0 contours)
Acoustic unit inventory
Language dependent resources

Fig. 1. The time and space-efficient architecture of the corpus-based TTS system.

2.1 Queuing mechanism used in the TTS architecture

An efficient queuing mechanism is implemented in the presented TTS architecture (Rojc & Kačič, 2007), where each double-linked list is used for one processing step in the TTS system. In this way all TTS processing steps are pipelined together. A queuing mechanism enables flexible addition and removal of dequeues from the mechanism, thus allowing for the merging of already existing processing steps, or adding new ones. The overall text-to-speech process runs in a loop, when processing the input text. All TTS engine dequeues are empty at the start. Firstly, the tokenizer module starts generating tokens from the input text by using a finite-state machine (FSM) based lexical scanner. Two additional token types are added for marking end-of-sentence or end-of-file conditions. These two tokens are only used for controlling the overall queuing mechanism. Immediately after detection, either of these two tokens, the following part-of-speech (POS) tagging dequeue is activated, taking all

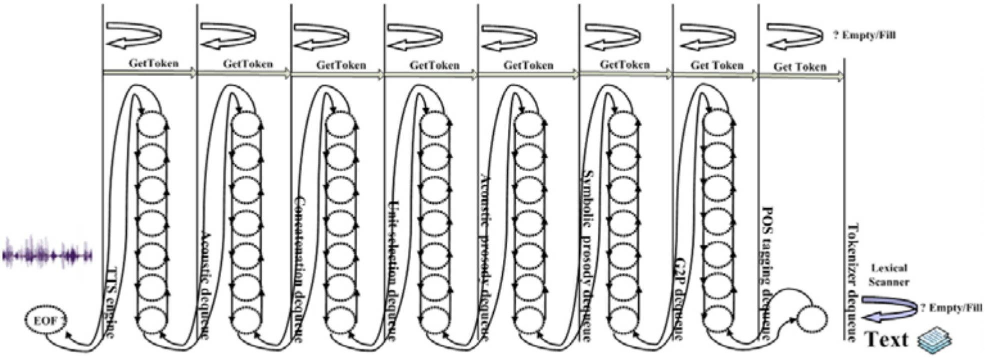


Fig. 2. The queuing mechanism.

tokens from the previous tokenizer dequeue (for the current sentence). After the tagging process, the grapheme-to-phoneme (G2P) conversion dequeue activates and grabs all tokens from the POS tagging dequeue. In this way (at the sentence level) the text-to-speech process continues until acoustic dequeue, where the speech signal for the corresponding sentence is finally generated. All text-to-speech processing steps are sequential processes. Nevertheless, the processing of several sentences within the presented queueing mechanism can run in parallel, by processing each sentence within its own thread. At the end, only the correct order from the input must be preserved, before playing-out generated speech signals.

2.2 Heterogeneous relation graphs used in the TTS architecture

All TTS processing steps contribute to the linguistic information used for generating the speech signal. The heterogeneous relation graph (HRG) structure provides clean general-purpose mechanisms for storing and representing all the information extracted by the TTS system (Rojc & Kačič, 2007; Taylor et al., 2001). In the PLATTOS TTS architecture, one HRG structure is used per each text sentence, and is accessible by all dequeues used in the TTS system. In this way, all algorithms are able to access, change, or enrich stored information when appropriate. Figure 3 illustrates the integration of the HRG structure into a queuing mechanism. The HRG structure demonstrates the use of two different relation-structures for storing extracted information, linear lists and trees. The linear lists are named Segment, Syllable, Word, Phrase, IntEvent, and SynUnits in Figure 3, whilst the tree structures are named SyllableStructure, PhraseStructure, IntonationStructure, SynUnitsStructure. The linguistic objects within the relation-structures are e.g. words, syllables, segments, phrase-breaks, intonation events, synthesis units, enriched with several attributes determined by the algorithms used within the processing dequeues. Attributes are the properties used in TTS system modules, e.g. part-of-speech, duration, phone-class and properties, intonation event type, phrase-break type, prominence label-type, to name just a few. Linear lists are used to specify the relation between linguistic items found in the specific processing step.

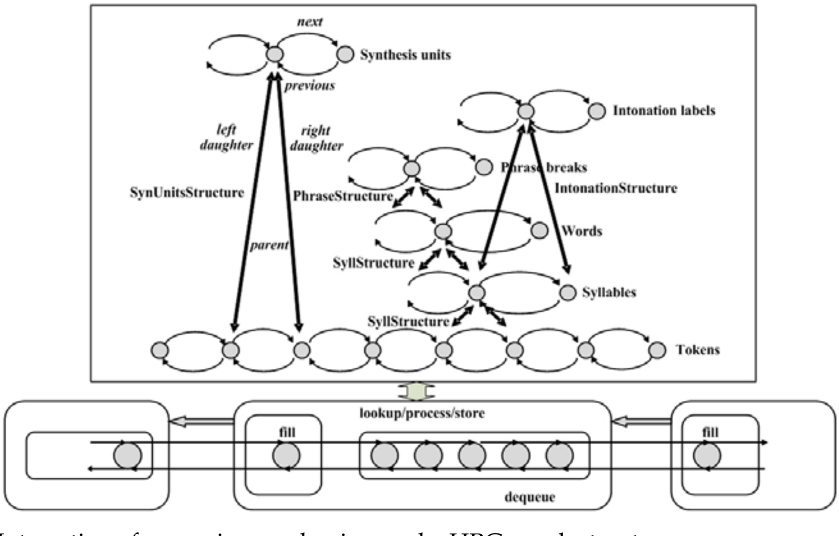


Fig. 3. Interaction of a queuing mechanism and a HRG graph structure.

Forward and backward traversals are possible within the structure. Additional tree relation-structures add vertical information between those linguistic objects included in different linear lists. In this way, very complex features for machine-trained models (e.g. CART trees, NNs etc.) can be generated from the linguistic information stored in the HRG structure, without any additional processing or extra work on feature construction. Furthermore, the relation-structures used within a HRG structure can easily be changed and adapted to different structures, following the processing needs of the modules used in the TTS system.

2.3 Finite-state machines used in TTS architecture

For multilingual and polyglot speech synthesis systems, it is important that the migration to new language can be done with little or no intervention in the algorithms used. This can be achieved by separating language-dependent language resources from the TTS engine, and obtaining a language-independent TTS engine. The efficient separation of language-dependent language resources is done within PLATTOS TTS architecture by finite-state machines (FSM) (Mohri, 1995; Rojc & Kačič, 2007). Furthermore, FSMs are also used for the representation of language resources and linguistic rules. FSMs can be constructed off-line and loaded into the TTS engine during on-line operation. The corresponding representation offers fast lookup, since the lookup does not depend on the size of the dictionary but only on the length of the considered input string. Minimization algorithms allow one to reduce the sizes of these devices to a minimum. The FSM compiler is used for the compilation of several regular expressions into the finite-state machine, construction of finite-state machine-based tokenizers, etc. In order to solve disambiguity problems, heuristically-defined or trained weights are assigned to FSM transitions and final states, yielding weighted finite-state automata and transducers (WFSA, WFST) that can be integrated into the TTS architecture (Mohri, 1995). In Figure 1 the tokenizer is marked as 'T' in the TTS architecture. At this processing level two-level rules or rewrite rules can be used, and compiled into finite-state machines by an FSM compiler (Mohri, 1996). Namely, these rules can resolve much of the language-dependent disambiguity present in the input texts. TTS system processes any given input text that often contains more or less spelling mistakes (e.g. e-mails, SMS messages). Therefore, the finite-state automaton 'S' follows (represents efficiently large lists of valid words), and is used by the spell-checking system (if it is included in the architecture). Using them, the spell-checking system is able to detect invalid words and can guess the most suitable replacements. Next, the POS-tagging module needs large-scale morphology lexicons. Therefore, the finite-state transducer 'P' can be used here for time and space-efficient representation of large-scale morphology lexicons. If TTS systems use rule-based POS-tagging algorithms (e.g. Brill, 1993), the POS-tagging rules can be further compiled into finite-state machines, and become a compact part of the TTS architecture (Emmanuel & Schabes, 1997). The grapheme-to-phoneme (G2P) conversion module uses, in general, large-scale phonetic lexicons for common words, proper names, and even foreign words, as found in the input text. All these resources can be represented by the finite-state transducer (FST) 'G', as presented in Figure 1. Decision-tree models can be included in the TTS architecture, since they represent efficient knowledge representation regarding time and space requirements. They can be used in the prosody modules (symbolic and acoustic prosody) for the prediction of phrase breaks, prominence and intonation event labels, segment durations, pauses between segments and the acoustic parameters of intonation events. Nevertheless, it has been shown that decision trees can also be represented by weighted finite-state machines (labelled as WFST 'SP', WFST 'AP') (Sproat & Riley, 1996). However, this step only makes sense when they are going to be merged with all other finite-state machines, as decision trees are already efficient knowledge-representation structures. In corpus-based TTS systems, the unit selection search process represents a significant time and space issue (large unit search space). Finite-state machines can be used here for more efficient access to unit candidates stored in the acoustic inventory. In the concatenation and acoustic modules, digital signal processing algorithms are used for the processing of concatenation points, and for adapting unit candidate pitch and duration and, in general, no external language-specific resources are needed.

3. Time and space efficient unit selection in corpus-based TTS systems

All the data-preparation steps needed for general corpus-based TTS systems are shown in Figure 4. The acoustic inventory and concatenation costs (have to be represented in a time and space efficient way) calculated between unit candidates are the result of these data preparation steps. The optimality and quality of the final acoustic inventory (used by the unit selection process) depends on several previous steps e.g. text-corpus construction, segmentation, phonetic tree-based clustering of units, and the acoustic evaluation of unit candidates. The last step takes care of removing acoustically similar units (the so-called redundant units) that are unnecessary in the optimised acoustic inventory. Calculation of concatenation costs then follows with a quantisation-based compression of these, and their space and time efficient representation, where the concatenation costs' matrix indices can be stored in the form of FSM.

3.1 Acoustic inventory construction

In corpus-based TTS systems the idea is to use the whole speech database for acoustic unit inventory, selecting the longest possible existing phonetic segments, and matching the target unit's specification, as defined for the target sentence. Because of the complexity and combinatorics of languages, it is important to find the best compromise: that has, on the one hand, as small a speech database as possible and, on the other hand, 'enough' acoustic realizations of those units found in several phonetic and prosodic contexts. Defining such a compromise is one of the major issues for the corpus-based speech synthesis approach (Bozkurt et al., 2003, Rojc, 2003). In PLATTOS TTS architecture diphone and triphone units are used within a unit-selection algorithm, where the diphones are base units. The richness of text corpus (regarding diphones and triphones) then has a significant impact on the richness of the acoustic inventory, on the performance of the unit-selection algorithm, and on the expected naturalness of the synthesised speech signal. A detailed analysis of several tokenised text corpora has to be performed in order to collect the appropriate text of a given language when striving to good acoustic inventory at the end.

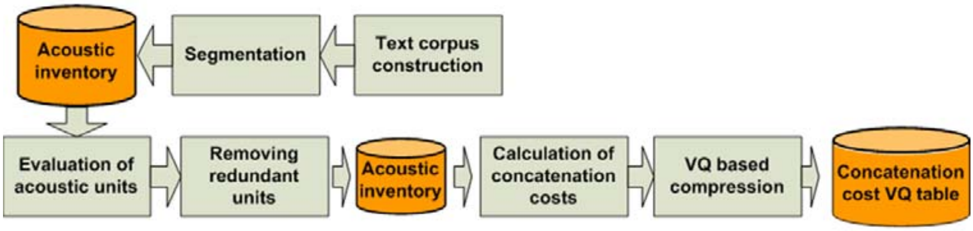


Fig. 4. Data preparation steps for the efficient unit selection process (off-line process).

After constructing a text corpus and recording speech database, the obtained database is segmented into unit candidates. Automatic segmentation procedures are preferred for segmentation of the database during the first step, but for optimal quality at least some manual checking usually follows. Better results can be expected when canonical phonetic transcriptions are verified and adapted to the recorded speech material, before running automatic segmentation. The final size of the constructed acoustic inventory is important in order to meet real-time requirements. In PLATTOS TTS architecture the starting acoustic inventory consists of a large set of non-uniform units (diphones and triphones). It can be expected that an acoustic inventory constructed directly from a segmented database, will contain acoustically similar units (can be qualified as redundant units) that can be removed. In order to detect these units, all units have to be acoustically evaluated regarding pitch, duration, and energy. When the text corpus is well-defined, the recorded speech material will more probably contain units with several distinct acoustical realisations, have less redundant units and, consequently, will allow for a better quality of synthesised speech from general input texts.

3.2 Acoustic inventory optimization

The search space for a unit-selection algorithm can already be reduced off-line during acoustic inventory construction, and also during the on-line unit selection process (within the TTS system) (Campbell & Black, 1996; Holzapfel, 2000). In the PLATTOS TTS system's unit selection approach, the reduction of the search space is proposed as a two-stage process (performed off-line). During the first stage, the diphone and triphone unit candidates are clustered according to their phonetic context and, during the second stage, acoustically similar units (similarity measurements are determined by considering pitch, duration, and energy) are automatically detected, and removed within the constructed tree-clusters. In order to detect acoustically similar candidates within the tree-clusters, detailed acoustic analysis is performed on all the cluster's unit candidates. Detecting acoustically similar units and removing them from the acoustic inventory can be performed in a manner analogous to the perceptual stimuli relationship. The final decision about which units should remain in the constructed clusters is done after the acoustic characteristics of all the clusters' unit candidates are obtained during analysis. The final optimised acoustic inventory contains, for each specified cluster, *n* units that are then used in the unit selection algorithm. The concept of suitability functions can be used in order to rank the unit candidates, where the setup and tuning of suitability functions can be performed by using a hybrid approach (Holzapfel & Campbell, 1998). First, the mean values of energy, pitch, and duration are calculated for each obtained tree cluster. All the mean values then represent those target values having a suitability value of 1.0. Other target values for energy, pitch, and duration, are defined for each unit candidate within a specific cluster by using the cluster's suitability functions' shape. Partial suitability functions must reflect acoustic differences between unit candidates within a specific cluster. Differences in duration amongst the units in the cluster 'i' are represented by using the following partial suitability function:

$$S[i]_{partial}^{dur} = \begin{cases} e^{-\frac{1}{2\cdot a^2}\cdot\left(\frac{dur-dur_{mean}}{dur}+a\right)^2}, & \frac{dur-dur_{mean}}{dur} < a \\ 1, \ -a \le \frac{dur-dur_{mean}}{dur} \le b & \\ e^{-\frac{1}{2\cdot b^2}\cdot\left(\frac{dur-dur_{mean}}{dur}-b\right)^2}, & \frac{dur-dur_{mean}}{dur} > b \end{cases} \tag{1}$$

Differences in pitch amongst the units in the cluster 'i' are calculated by using the following partial suitability function:

$$S\left[i\right]_{partial}^{f0} = \begin{cases} e^{-\frac{1}{2 \cdot c^2} \cdot \left(\frac{f_0 - f_{mean}}{f_0}\right)^2}, & \mathrm{f}_0 - f_{mean} < 0 \\ e^{-\frac{1}{2 \cdot d^2} \cdot \left(\frac{f_0 - f_{mean}}{f_0}\right)^2}, & \mathrm{f}_0 - f_{mean} \geq 0 \end{cases} \tag{2}$$

Differences in energy amongst the units in the cluster 'i' are calculated by using the following partial suitability function:

$$S\left[i\right]_{partial}^{en} = \begin{cases} 1, & \mathrm{en} - en_{mean} > 0 \\ e^{-\frac{1}{2 \cdot e^2} \cdot \left(\frac{en - en_{mean}}{en}\right)^2}, & \text{en-en}_{\text{mean}} \leq 0 \end{cases} \tag{3}$$

In this way, the overall unit candidate's suitability is ultimately defined by combining partial suitability functions for pitch, duration and energy within the given cluster:

$$S_{overall} = \prod_{i=1}^{N} S_{partial} \tag{4}$$

At the end of the ranking process unit candidates are ranked within the region of 0 to 1 in all tree clusters. This region is then divided into smaller sub-regions. Suitability values within a specific sub-region correspond to those candidates that have similar acoustic characteristics, meaning that they have small or insignificant differences regarding pitch, duration, and energy. In this case, only one candidate from a specific sub-region is kept in the optimised acoustic inventory, and all the others can be removed. Multiplication of the used partial suitability functions ensures that differences in certain acoustic parameter are noticeable within the overall suitability value for each unit candidate, and that the significance of a particular acoustic feature is reflected by the shape of the used partial suitability function.

3.3 Concatenation cost calculation and representation

The calculation of concatenation costs (CC) is a very time-consuming step for corpus-based TTS systems, especially if performed during the on-line unit selection process. Namely, the CC costs must be calculated between all phonetically-matched candidates for each of the two successive target units in the current sentence. Then, in order to evaluate any distortions at concatenation points, the corresponding speech samples of all these candidates have also to be loaded. In order to avoid this, the obvious solution can be the off-line calculation of all CC costs. The disadvantage of this solution is that the target unit sequences are unknown and, therefore, consideration of any phonetically-matched candidates in the acoustic inventory must be taken into account. Furthermore, concatenation costs have to be calculated between all unit pairs in the acoustic inventory (for large databases, non-uniform acoustic inventories can have a lot of units), and this results in large CC cost-matrix dimensions and storage requirements. In order to also solve this problem, the vector quantisation algorithm (VQ) can be used. By using the VQ technique, we are able to compress a CC cost-matrix into a much smaller one. This whole process can be easier performed by first splitting the large CC cost-matrix into smaller sub-matrices (speed and memory problems). The CC costs for each pair of candidates are calculated for each sub-matrix. The calculated costs within each sub-matrix are then quantisized into a corresponding codebook of a pre-defined size (number of clusters) (Rojc & Kačič, 2007). Without using vector quantisation, the storage requirements for each sub-matrix are (W – size of the sub-matrix):

$$Storage = N \cdot W \cdot sizeof(float) \tag{5}$$

After using vector quantisation, the storage requirements drop down to:

$$Storage = VQ \cdot W \cdot sizeof(float) \tag{6}$$

VQ represents the codebook dimension regarding a pre-defined size for any specific sub-matrix. After calculating codebooks for all sub-matrices, they are merged into one common codebook, representing the compressed CC cost matrix. This representation is a space-efficient representation of concatenation costs between all candidates in the acoustic inventory. An index table is also built in addition to the constructed codebook, and used for accessing the concatenation costs. In order to also make the CC cost lookup also time and space-efficient, the CC cost indices are stored in the form of a finite-state transducer (FST) (Mohri, 1997). In this way we are able to perform an efficient lookup process in the unit selection algorithm.

3.4 On-line unit selection algorithm

The unit selection algorithm is a very important process in corpus-based concatenative speech synthesis, since it searches for the best matching sequence of unit candidates with those target units specified for the input sentence. The selection of non-uniform units (diphones and triphones) from the acoustic inventory is based on minimising those acoustic distortions that originate from concatenations, and minimising the needed modifications of the unit candidates. In the PLATTOS TTS architecture, these distortions are described in the form of two costs:

- target cost $C^t(u_i, t_i)$: represents an estimation of the difference between unit candidate u_i in the acoustic inventory and target unit specification t_i,
- concatenation cost $C^c(u_{i-1}, u_i)$: represents an estimation of the quality of the concatenation of two successive units u_{i-1} and u_i.

Target unit specifications include e. g. phonetic symbol, symbolic prosody information (e.g. stress indication), acoustic prosody information (e.g., desired unit duration and F0) etc. They are used for calculating target cost (TC) in the on-line unit selection algorithm. Concatenation cost (CC) is already calculated off-line (as suggested), and accessed through an efficient lookup process. Common cost then reflects the differences in target and acoustic realisations for specific unit candidates, and the expected distortions when the selected unit candidates are concatenated together. In corpus-based TTS systems, a pitch and duration modification algorithm (e.g. TD-PSOLA) is often applied to pre-stored candidates in the acoustic inventory, in order to guarantee that the prosodic features of synthetic speech meet the predicted target values. Then, using a good criterion for finding the best fitting unit sequence from the acoustic inventory is crucial for generating high quality speech. In the PLATTOS TTS architecture unit selection algorithm, the following equation is used for calculating the common cost for each unit candidate:

$$C(u_i) = C^t(u_i, t_i) + C^c(u_{i-1}, u_i)$$
$$C(u_i) = w_{unit} \cdot \left[\prod_{j=1}^{P} S_{partial} \right] + w_{unit} \cdot w_c \cdot \left[(S_{conc})^{w_{local}} \right] \quad (7)$$

It follows from equation (7), that the target cost uses partial-suitability functions for duration, pitch, and energy (equations (1), (2) and (3)). The mathematical framework behind the computation of this common suitability is based on fuzzy-logic (Holzapfel & Campbell, 1998). The performance of the unit selection algorithm and, consequentially, the quality of the synthesised speech, significantly depends on the partial suitability functions' parameters. Furthermore, weight w_{local} is additionally included in order to have control over the calculated concatenation cost between two unit candidates. Weight w_{unit} is included in order to favour the selection of longer units during the unit selection process (e.g. triphones, instead of diphones). Finally, weight w_c is included in order to control the influence of concatenation cost on the common cost $C(u_i)$. And S_{conc} represents the distortion measure between two successive unit candidates, based on an acoustic cost that is calculated by using signal processing based on spectral analysis (suggested to be performed offline).

3.5 Gradient-descent based unit selection process optimization

An important common cost calculation issue is the optimal setup and tuning of those parameters used within partial-suitability functions (equations (1), (2) and (3)), and other weights used in equation (7). Parameters a, b, c, d, and e influence the shapes of the partial-suitability functions and, consequently, influence the significance of a particular criterion (duration, pitch, energy) within the unit selection process. Furthermore, searching for the best unit sequence using a unit selection algorithm is a multidimensional problem. Heuristics is usually used for setting up parameters and weights, or extensive subjective listening tests are performed, resulting in more or less optimal solutions. Such an approach is at least time consuming and laborious. Besides, parameters and weights have to be adapted for each new TTS voice. Instead, the PLATTOS TTS system uses an automatic optimisation approach of cost function's weights based on a relaxed gradient descent algorithm (RGD) (Figure 5).

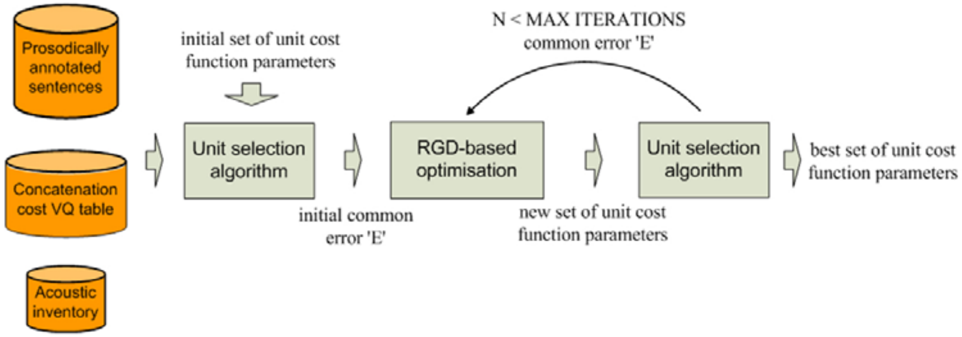

Prosodically annotated sentences
Concatenation cost VQ table
Acoustic inventory
initial set of unit cost function parameters
N < MAX ITERATIONS
common error 'E'
Unit selection algorithm
RGD-based optimisation
Unit selection algorithm
best set of unit cost function parameters
initial common error 'E'
new set of unit cost function parameters

Fig. 5. Unit-selection optimisation process.

The input into the unit-selection optimisation process is a set of prosodically annotated sentences (HRG utterance structures), off-line calculated CC costs, and acoustic inventory. The unit selection algorithm then selects a sequence of units for each input sentence by using an initial setup of weights (the initial setup of values is set by a hybrid approach, as proposed in (Holzapfel & Campbell, 1998)). Automatic unit selection process evaluation is performed during the next step, by calculating the pitch deviations of neighbouring selected candidates, and deviations between selected candidates' durations and those durations predicted by prosody. The obtained evaluation result represents initial common error 'E' for the unit selection optimisation process. The process then keeps running within a loop, and the weights and parameters are iteratively updated by using the RGD technique. The optimisation loop consists of several processing steps. During each iteration, a common error 'E' is calculated as the sum of pitch differences (between predicted pitch $\hat{f}_0(k)$ and the selected unit candidate's pitch $f_0(k)$), and the duration differences (between predicted duration $\hat{d}(k)$ and the selected unit candidate's duration $d(k)$):

$$E = \sum_{k=1}^{N} \left[e(k) \right]^2 \quad , \quad e(k) = \frac{\left| \hat{f}_0(k) - f_0(k) \right|}{\hat{f}_0(k)} + \frac{\left| \hat{d}(k) - d(k) \right|}{\hat{d}(k)} \tag{8}$$

The computation of common error 'E' includes differences in durations (time) and F0 values (frequency). Therefore, differences in duration and F0 value are normalized in equation (8). In order to minimize these differences (and common error 'E'), the RGD technique is used, optimizing the initial setup of the unit-cost functions' weights and parameters. In other words, the goal is to minimize the cumulative common error 'E'. All weights and parameters are stored in a vector **p**:

$$p = \left[p_1, p_2, ..., p_l \right] \tag{9}$$

This vector is then iteratively updated in such a direction that the change results in a smaller common error 'E'. This direction is searched for by a gradient calculation, performed for each value in vector **p**: $\nabla E(p)$. An adjustment of each vector value is then performed by the following update rule:

$$p_{n+1} = p_n - diag(\mu_n) \cdot \nabla E(p_n) \tag{10}$$

The obtained gradient vector consists of the partial derivatives of the unit cost functions, with respect to each value of the vector **p** (e.g. partial derivative of 'E' with respect to p_1):

$$\frac{\partial E}{\partial p_1} = 2 \cdot \sum_{k=1}^{N} e(k) \cdot \frac{\partial e(k)}{\partial p_1} \tag{11}$$

The adaptation rate μ must be selected so that the convergence of the algorithm is guaranteed, since the performance of the algorithm is quite sensitive to a proper setting of the adaptation rate. Namely, if the adaptation rate is too high, the algorithm may oscillate and become unstable. On the other hand, if the adaptation rate is too low, the algorithm will take too long to converge. The optimisation process is repeated, until obtaining the predefined minimal error. When this happens, the set of weights and parameters is stored, and can be used for the on-line unit selection algorithm in the corpus-based TTS system. The approach is fully automatic, and can be repeated for each new voice used in the TTS system.

4. Personification of the TTS systems

The idea of advanced human-machine interfaces and spoken dialogue systems is to emulate natural and highly-complex human-human interactions. Substantial effort by several researchers has already been devoted to this task, by taking into account multimodal-input and multimodal-output contexts. An understanding of attitude, emotion, together with how gestures (facial and hand) and body movement complements, or in some cases even overrides verbal information, provides crucial information about modelling interactive management. It influences both input and output perspectives of the realization of natural human-machine interaction and, consequently, the personification of those TTS systems used in e.g. spoken dialog systems. Personification of TTS systems, therefore, not only relates to the transformation of a TTS system's output into ECA's visually-presented articulation within the mouth region (visualizing verbal behaviour), but also to the visualization of non-verbal behaviour. The most natural way to visualize (emulate face-to-face conversation) both verbal and non-verbal information is to translate it into a human-body representation. Embodied conversational agents (ECA's) are widely used concepts for the visualization of conversation and are used in many spoken dialogue systems. ECA implementations range from talking heads (Poggi et al, 2005), to agents that can move and use the whole representation of the human body (Heloir & Kipp, 2009; Thiebaux et al., 2008). There are many implementations of ECA's that can, in one way or another, emulate natural human behaviour and evoke emotional and social responses within human-machine dialogue. (Ball & Breese, 2000) describe the generation of emotional responses and the recognition of emotions by humans, and the additional adaption of ECA's personality to that of the human. (Poggi &Pelachaud, 2000) generate communicative behaviours on the basis of speech acts and concentrate on one facial expression and speech act performatives. Performatives are a key part of the communicative intent of a speaker, along with propositional and interactional acts. In general terms, all conversational behaviour in conversational models must support conversational functions and different input/output modalities. Any conversational action in any modality can result in several (sometimes contradictive) communicative goals. The general architecture of a system that can visualize and personificate a general TTS system, used in e.g. a spoken dialog system, is formed as shown in Figure 6.

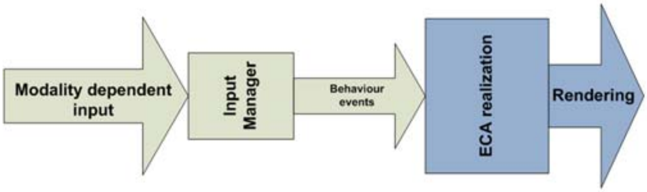


Fig. 6. General architecture of an ECA visualization system.

The idea of visualizing suggests that different input modalities are combined into different behavioural events. Different input modalities are commonly generated as abstract behaviour descriptions provided in XML based description schemes such as Affective Presentation Mark-up Language (APML) (DeCarolis et al., 2004), and behavioural mark-up language (BML) (Vilhjalmsson et al., 2007). These are general description languages that can be used to describe any movement/action realized within the scope of human-machine dialogue. The task of an input manager, commonly referred to as a behaviour modeller, is to process different modality-dependent inputs, and to transform them into a time-referenced set of behavioural events. The key concern of such a time-scheduling process is to synchronize verbal with non-verbal behaviour, such as facial expressions, head movements, gaze and head gestures. Such behaviour often relies on the semantic information of data, such as non-standard sync-points at word breaks, dialogue markers, etc. The behavioural events (behaviour controllers) then form speech-synchronized descriptions of motion that should be transformed into movement on an ECA's articulated model (body). Different types of articulated bodies can be used (Güdükbay et al., 2008). In general 3D models can be grouped into:

- *Stick figure models*: models based on sets of rigid elements, and connected to joint chains.
- *Surface models* (mesh-based models): represent an upgrade of stick figure models. In this case, a polygonal mesh-layer (skin) is applied on the skeleton chains.
- *Volumetric models*: use simple volumetric primitives such as spheres, cylinders and ellipsoids, in order to construct the body shape.
- *Multilayered models* (muscle-based models): present anatomically-correct models. The animator of such models introduces different kinds of constraints to the relationship between layers.

The ECA realization engine (Figure 6) is used to store the articulated models of different ECA's (different bodies), and to apply behavioural events in the form of different transformations on the control units (parts of the articulated model used to generate movement). These transformations result in animated movement. The type of animation technique used depends on the type of articulated model. Most commonly, such animations are performed in the form of skeletal joint transformations and morphed-shaped transformations. The proprietary EVA framework (Mlakar & Rojc, 2011), developed to evoke a social response in human-machine interaction, is a python-based software environment that can convert a TTS system's output into audio-synchronized animated sequences of speech. ECA's provided by EVA framework can generate social responses in the form of facial expressions, gaze, head and hand movement and, most importantly, in the visual form of synthesized speech. The EVA framework provides a description script, an animation engine and articulated 3D models, and provides visual representation of synthesized speech sequences in the form of different types of video streams (in addition to synthesis into a video file/screen). Figure 7 outlines the modular architecture of the EVA framework.

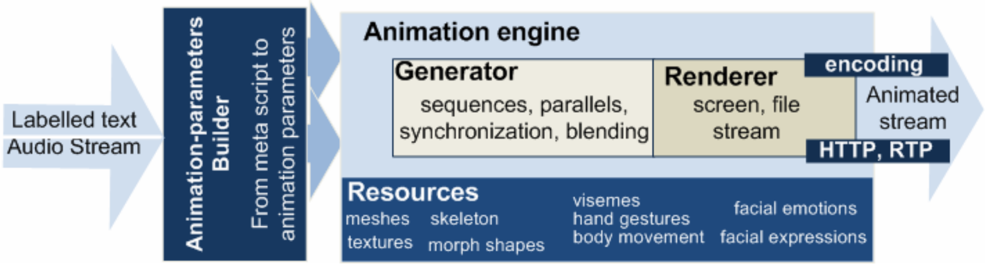


Fig. 7. Architecture of the EVA framework.

In order to personificate a TTS system by using the EVA framework, the TTS system has to produce TTS output according to the framework's specifications, based on the EVA Script XML scheme (Mlakar & Rojc, 2011). These XML schemes specify the desired ECA facial animations and body movements. The animation engine translates them onto the articulated 3D model of a human body in the form of animated movement. A TTS system's output can contain acoustic, linguistic, syntactic, semantic, and temporal information about general input texts that can be realized within a two stage visualization concept. The first stage is called 'Animation building' and the second 'Animation realization'. The Animation building stage transforms TTS output in the form of the EVA script XML scheme into animation parameters mapped to different control units of the ECA's 3D articulated model. The transformation from abstract to animatable content is then performed by the Animation parameter's builder. Such a transformation can be described as interfacing different XML tags using different ECA resources. Each ECA generated by the EVA framework has two types of resources. The 3D multi-part actor resources (Figure 7) contain different 3D-submodels of body (e.g. hair-style, eyes, teeth, dresses, etc.). Each 3D-submodel is associated with its corresponding textures, polygonal meshes and sets of different control units (morphed shapes, skeletal chains). The Personality template resources contain different behavioural templates written in the EVA script. These templates describe the common articulation of an ECA (e.g. how should, in general, specific viseme be formed), triggering words to gesture translations (e.g. what is a common gestural sequence when a certain word occurs), and other distinctive features of an ECA (e.g. eye-blinks, probability of gesturing, etc.) making each ECA an individual 'person'. The Animation parameters builder, therefore, translates the labelled text by interfacing each EVA script tag with a control unit, or behavioural template, and forms different groups of movements. Each group of movement is defined by semantic (which control units in which order), temporal (the duration of stroke, hold, and retraction phases) and spatial features (ending position of the control unit). The Animation realization phase transforms animation parameters into animated sequences. The animation parameters present raw data that describes how the Animation engine should move different control units. The Animation engine of the EVA framework takes care of animating and rendering the obtained animation parameter sets. It is based on the Panda 3D game engine (Goslin & Mine, 2004). In essence, this animation engine transforms the animation parameter sets into corresponding sequential and/or parallel movements of control-points (bones, or morphed shapes) lerp intervals. Each control-point in 3D space can be moved, either by 3D transitional or 3D rotational vectors (as specified in MPEG4 standard). The Forward kinematics and animation engine's generator provide procedures for the synchronization of such movements, and implement the animation-blending technique used on those animated segments that have to be controlled by different gestures at the same time (e.g. both smile and viseme can try to control the lower jaw joint; in such cases most of the influence is given to the viseme, and only a small portion is left to the facial gesture smile). Based on the semantics of the animation parameter sets, the animation groups the control units into sets of sequential and concurring movements and associates each movement with its temporal and spatial features, therefore forming personalized body movements. The EVA framework also presumes that no movement is linear and should, therefore, be interpolated against its interpolation curve. The EVA framework provides three types of non-linear interpolation for each personalized movement: EaseIn (slow-start and ramp-to-full, abrupt finish), EaseOut (starts with full speed and in the last n frames decelerates to a slow stop), and EaseInOut (starts slowly, ramps to full speed and after the constant phase, if it exists, slowly decelerates to full stop). The rendering process is frame based and at each frame is interpolated against its non-linear interpolation curve. At any given frame the animation can also be stopped/paused or re-adjusted to its given temporal/spatial features. The EVA scripts describe both verbal and non-verbal behavior, independently. The verbal behaviour is contained within speech XML tags named speech, and the non-verbal behaviour may be contained within fgesture and bgesture tags describing facial expressions, and different body gestures. The verbal parameters can be described by the semantic, temporal, and articulation features of a sequence, whereas non-verbal behaviour involves describing the presence level of facial expressions and different body gestures. All speech-driven non-verbal behaviour can be defined in the TTS system's output directly, or indirectly by derivation of several non-verbal parameters found in the TTS generated output. Non-verbal parameters, such as emphasis, phrase/word breaks, and key phrases (e.g. dialogue discourse markers) are used when the non-verbal behaviour is controlled by a TTS system. The non-verbal feature allocators, fgesture and bgesture unify a set of control units, assigned to control different parts of the body. The facial expressions contain control units that can be physically assigned to the human face (e.g. control units such as lower-jaw, mouth corners, etc.). Similarly, body gestures allocate control units, such as: left elbow, neck, control units for fingers, etc. In addition, the left and right-eye control units are also assigned to the body gesture group. By describing the temporal and spatial features of movement in the form of sequential or parallel groups, the EVA framework enables hierarchical levels of animation for both fgesture and bgesture objects. Each movement can, therefore, be built from different control units with either sequential or concurrent movement. In the context of spoken dialogue systems using TTS systems, the EVA framework not only enables more realistic human-machine interaction, but can also evoke emotional and social responses that exist in face-to-face human-human spoken dialogues.

5. Multilingual and multimodal PLATTOS TTS system for the Slovenian language

This section presents an implementation of the corpus-based PLATTOS TTS system for the Slovenian language, using a concatenative approach and a TD-PSOLA speech-synthesis algorithm. The dequeues are tied together into a common time and space-efficient TTS engine, using the HRG structure for the representation of linguistic information. Finite-state machines, however, are used for efficient language resource representation, and separation of the language-dependent part from the language-independent TTS engine. The fsmHal library is used to efficiently construct the necessary finite-state machines used (Rojc, 2000; Rojc 2003). All modules, as specified in the TTS system architecture (Figure 1), are included and used. In the following subsections, implementation of those modules used for the personalized PLATTOS TTS system regarding the Slovenian language is presented in detail.

5.1 Tokenizer dequeue

All tokens are specified off-line in the form of regular expressions. Then the FSM compiler is used for the construction of a tokenizer finite-state machine. The additional part of the tokenizer module is the spell checker. It is used in order to prevent erroneous words that corrupt the performance of other modules within the TTS system, e.g. obtained prosody patterns result in speech signals with lower intelligibility. The spell-checking algorithm uses a large word list, containing a set of valid words. Represented as FSA, the corresponding list is then used for edit distance calculations and searching for the best possible replacements for the misspelled words found in the input text. An additional part of the tokenizer is the normalisation process. Number tokens' factorization is performed firstly, in order to convert the numbers into the corresponding word forms. Some languages (e.g. German and Slovene) need additional filter (FST), for handling language-specific decade flop phenomenon. The core number lexicon is constructed from the SIlex lexicon (Rojc & Kačič, 2000), represented as FST. Additional rewrite rules are used for language-specific word insertions (special words such as ''and'' (English), ''und'' (German) or ''in/and'' (Slovene). Compiling rewrite rules into a FST is performed, since it is more efficient and requires a limited number of operations (Sproat, 1998). Furthermore, an important issue is the normalisation process of abbreviations, especially in the cases of inflectional languages. When considering the context a decision has to be made about which conversions are possible and which are impossible. The marking of unacceptable and acceptable conversions for a given context is done using the rewrite rules, and written by an expert. For processing special symbols (e.g. %), the construction of FSM representing lexical analysis for a given symbol, is performed for the conversion of a special symbol into word forms. In those cases where more possible conversions are preserved at the end, the most appropriate one is obtained using the BestPath algorithm (Rojc & Kačič, 2007).

5.2 POS-tagging dequeue

The POS tagging approach performed in the PLATTOS TTS system is based on the Brill's POS tagging approach. The POS tagging process consists of several steps. Firstly, the morphology lexicon is used as obtained from the training. Within this lexicon each entry is assigned the most probable POS tag found in the training corpus. If a word is not found, the SImlex morphology lexicon (Rojc & Kačič, 2000) is used next. Deterministic and minimized FST representation of SImlex lexicon represents time and space-efficient representation and fast lookup time. Morphological analysis then follows, which uses the so-called guessing automata, constructed for unknown words (this FSA tries to guess the POS tag by analysing word endings) (Daciuk, 1998). POS tagging context rules are used at the end. Within the scope of the post-processing stage, local grammars are used to resolve possible remaining ambiguities, e.g. as a consequence of systematic tagging errors that are unsolved during the POS-tagging process (Rojc & Kačič, 2007).

5.3 Grapheme-to-phoneme conversion dequeue

The SIflex phonetic lexicon for common words is first used during the unified approach to grapheme-to-phoneme conversion (Rojc & Kačič, 2007). Additionally, the SIplex phonetic lexicon for proper names is included, followed by the homograph detection step. Next, possible unknown words are converted into phonetic transcription by using trained CART tree models for stress, grapheme-to-phoneme, and syllable prediction. The HRG utterance structure is used as a linguistic knowledge source and for feature construction. Therefore, several complex features can be easily constructed by using a textual list of the linguistically attributed names. Syllable markers are also inserted into phonetic transcriptions (in the case of unknown words), since this information is important later for prosody modules. In the final stage of the unified G2P process, several rules have to be applied for performing the post-processing of the canonical phonetic transcriptions, by also considering cross-word contexts. Namely, in the Slovenian language cross-word context has a significant impact on pronunciation and must be considered within the whole G2P conversion process. The expert defines these rules for all phoneme conversions, occurring at word beginnings and word endings. Furthermore, input texts often contain words or phrases from some other language. The first problem that has to be solved is to detect such words in efficient ways, and the second is to specify the corresponding pronunciations. When e.g. the Slovenian input sentence contains a German name "Gerhard Schröder'', these two words have to be detected, and then converted into the phonetic transcriptions using Slovenian phonemes. As suggested in (Rojc & Kačič, 2007), a G2P conversion module for the German language (using SIplex lexicon) is used first, and then German phonemes are mapped into the most suitable phoneme substitutions defined for the native language. This mapping can be done by using the phoneme mapping table constructed by the phonetic experts. This polyglot functionality is currently supported for German and English names.

5.4 Symbolic and acoustic prosody dequeues

Within symbolic prosody module the prediction of phrase breaks, prominence labels, and Tilt intonation labels (based on syllable level) is performed (Taylor, 2000). CART trees are used, since classification is performed on several discrete linguistic attributes during training. The phrase break prediction model inserts phrase break labels, the prominence prediction model marks the prominent syllables, and the intonation prediction model assigns Tilt intonation labels to each syllable. In the PLATTOS TTS system for the Slovenian language, a B3 label is used for labelling major phrase breaks, and a B2 label for minor phrase breaks. Additionally, phrase break positions are used for pause insertions in the sentence. Prominence labels on syllables are marked as PA (primary accent, assigned to the most accentuated syllables inside the intonation prosodic phrase), and as NA (marking secondary accents in the prosodic phrase). Tilt intonation event labels (a c l m fb rb afb arb lfb mrb mfb lrb) are assigned to each syllable in the sentence. In the acoustic prosody module prediction is performed for segment durations, pause durations at phrase break positions, and the prediction of Tilt acoustic parameters for each Tilt intonation event. Here, regression trees are used because of the nature of the used data. Separate prediction models are used for vowel phoneme duration prediction, and for the prediction of consonant phoneme durations. An additional tree model is trained for the prediction of pause durations, using only sentence internal pauses in the recorded Slovenian speech database (female voice). After the Tilt acoustic parameters have been predicted, reconstruction of the specified F0 contour can also be performed (for subsequent modules), and is stored within the HRG structure (Rojc & Kačič, 2007).

5.5 Unit selection dequeue

The input text corpus (newspapers, literature, internet) used for recording the Slovenian speech database consists of approximately 31 million words. The main criteria for selecting sentences were: richness with different diphone and triphone units, maximal final size of the speech database, and the minimal and maximal lengths of the sentences (Rojc, 2003). Before the segmentation process of the database into monophone, diphone and triphone units, the canonical phonetic transcriptions were manually verified, and adapted to the recorded material of the database. The initial acoustic inventory was constructed from a large set of non-uniform units (diphones and triphones). Then, the two-stage search space reduction process was performed, as presented in section 3.2. During the first stage, all diphones and triphones were clustered by considering the phonetic contexts of the units (by using a tree-based clustering technique). The constructed trees were used in the next stage - in the process of eliminating acoustically-similar unit candidates (redundant units). Since calculation of concatenation costs is a very time consuming process, they are calculated off-line, as presented in 3.3, in order to achieve better time and space efficiency of the on-line unit selection process. A vector quantisation algorithm (VQ) was also used in order to minimize the huge CC cost matrix. The final VQ codebook and the FST with CC cost indices then enable an efficient CC cost lookup process in the on-line unit selection dequeue. As already mentioned, at the end of symbolic and prosodic dequeue, the sequence of target units with predicted symbolic and acoustic prosodic parameters is already defined. The next step is then to search in the optimised acoustic inventory for the best matching unit candidates. The basic strategy in the PLATTOS TTS system is to find the longest non-uniform unit for each target, ensuring also that the acoustic features and phonetic contexts of the unit candidates are as close as possible to the target unit specifications (stored as HRG utterance structure).

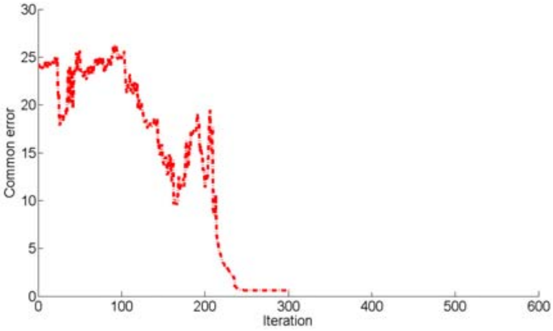


Fig. 8. Common error ‘E’ during the unit selection optimisation process.

As presented in 3.5, an important issue for an on-line unit selection process based on common unit cost calculation is the proper setup and tuning of partial suitability functions’ parameters and weights, as used in equation (7). The tuning of these parameters and weights is performed from non-optimised acoustic inventory containing all database diphone and triphone units, together with the off-line calculated concatenation costs. The initial weights and parameters are defined by using the hybrid approach. The input into the unit selection optimisation process is a set of prosodically-annotated database utterances (100 sentences). When considering the defined prosody and the existing unit candidates in the acoustic inventory for each sentence unit, the optimisation process iteratively searches for a sequence of such units that would result in minimal F0 mismatches between selected candidates, and in minimal duration deviations towards the predicted prosody. After each iteration, the RGD algorithm evaluates the common error ‘E’ made by selecton of candidates (regarding pitch and duration), and updates parameters and weights before the next unit selection process occurs. Common error ‘E’ distribution across all iterations for the set of database utterances (female voice) is presented in Figure 8. The iteration having the smallest common error ‘E’ specifies the proper set of weights and parameters to be used in the on-line unit selection algorithm. As can be seen, the RGD algorithm does not stop if the common error ‘E’ starts to increase, in order to avoid local minima - otherwise the unit cost functions’ weights would have been already specified at visible minimum found after 30 iterations. The optimised acoustic inventory, tuned unit cost functions’ parameters, and weights are then used in the on-line unit selection algorithm. The best sequence of unit candidates is found by using finite-state machine based BestPath algorithm that minimises the two costs along the input sentence: target cost and concatenation cost. The unit selection algorithm significantly reduces the amount of needed signal processing in order to meet the predicted prosody characteristics at the end, which naturally improves the quality of the generated speech. Figure 9 shows the common error 'E' per sentence (in the set of 100 sentences). It is composed of F0 mismatches between selected candidates, and of selected units’ duration deviations towards unit target specifications, defined by both prosody modules. All errors are summed within each sentence and then divided by the number of selected units in the sentence. Therefore, the normalised common error values are actually presented, in order to compare the obtained values between sentences in the given test set. It can be seen that the common error 'E' across the whole set of sentences is significantly larger when running a non-optimised unit selection algorithm (x markers), and that the common errors 'E' in the case of the optimised unit selection algorithm are smaller (circle markers).

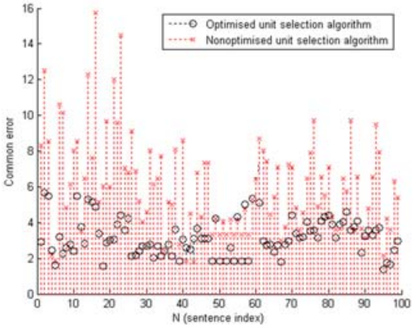


Fig. 9. Common errors ‘E’ on the set of test sentences (100 sentences).

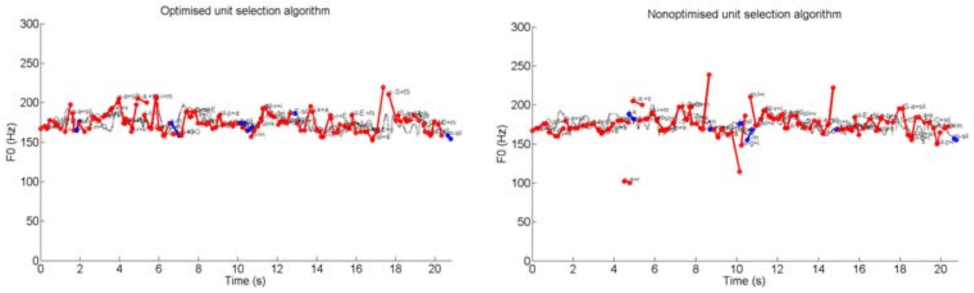


Fig. 10. Selected unit candidates when using (a) optimised cost functions’ weights, and (b) non-optimised cost functions’ weights.

Further, the sequences of the best candidates selected by using optimised and non-optimised unit functions’ weights and parameters are shown in Figure 10. Here, each selected unit candidate (diphone/triphone) is represented by a straight line. The length represents the duration of the selected unit candidate. The start and end-points of each line are characterized by the F0 values at the start and end of each unit candidate selected from the acoustic inventory. Naturally, the goal of the unit selection process is that observed F0 differences between successive straight lines are as small as possible, and that pitch values between candidates are also as close as possible. Namely, this will result in lesser-needed signal post-processing to be performed by TD-PSOLA. When comparing figures (a) and (b), it can be seen that by using an optimised unit cost functions' weights and parameters, the F0 mismatches are smaller (and F0 points between successive units are closer), which results in a more fluent and more natural synthesized speech signal.

5.6 Concatenation and acoustic dequeue

The concatenation module processes those units selected by the unit selection process in the previous dequeue. Since the following acoustic module is based on the TD-PSOLA algorithm, this module takes care of the following processing steps: calculation of analysis pitches, searching for an optimal concatenation point between two successive units, matching of analysis and synthesis pitches, and the smoothing of concatenation points. The acoustic module based on the TD-PSOLA algorithm is then used for changing the durations and pitch on those selected units, where existing F0 mismatches and duration deviations are unacceptable (Rojc & Kačič, 2007).

5.7 PLATTOS ECA – EVA –

ECA EVA (Mlakar & Rojc, 2011) is a PLATTOS TTS system's conversational agent that can be used in different spoken dialog systems. ECA EVA represents the personification of the PLATTOS TTS system, implemented by using the EVA framework. The PLATTOS TTS system output's synthesised speech and linguistic and acoustic data of the input sentence (contained in the HRG structure) in the format of the EVA XML script. Currently, these scripts contain sequences of phonemes, visemes, and gesture triggers. Each generated EVA script includes corresponding temporal (duration), and spatial information (e. g. articulation). By using this input, the EVA framework is then able to visualize a PLATTOS TTS system's output in the form of animated verbal and rule-based non-verbal behavioural response. ECA EVA is a female agent, since the selected ECA gender depends on of the voice used in the TTS system. It can synthesize expressive speech sequences based on different levels of co-articulation, head and hand gestures, facial expressions and emotions, and gaze. The lip-sync process synthesizes verbal features, and employs the articulation parameter (stress) at spoken sequence and utterance levels. At the spoken sequence level of articulation all utterances are additionally modified to meet the general articulation properties of the sequence as a whole. Articulation at the utterance level only modifies the spatial properties of the selected utterance. In this way, spoken dialogue-flow can not only adapt articulation, but also influence the speed at which a certain answer is spoken. Therefore, in addition to articulation relating verbal features, the general articulation can also define several personality features of an ECA (e.g. fast-speaker, speaker with good articulation etc.). If, for instance, the user did not understand the different parts of the spoken sequences, such sequences can be repeated at a slower rate and with a higher level of articulation. Therefore, such a personificated PLATTOS TTS system can also be used as a tool for learning pronunciations and other learning/entertainment applications. The rule-based non-verbal behaviour, such as: emotion, facial expressions, head and hand movements, are generated based on linguistic and acoustic information (stored in the HRG structures) that the PLATTOS TTS system can currently provide (e.g. morphology information, phrase-break labels, prominence labels, trigger words and phrases, stress levels, pitch etc.). By using emphasis markers word/phrase-break markers, ECA EVA can generate different speech-driven pointing gestures that can visually emphasize a certain word/phrase. By interfacing words/phrases with different emotions, and facial expressions, EVA can visually generate speech-driven facial expression, such as: speaking with a gentle smile, saying something sadly, etc. All these features represent an essential part of the visual synthesis from TTS output. Figure 11 demonstrates the personification of the PLATTOS TTS system including expressiveness and emotions that are already well-supported by the EVA framework. Different speech segments can be accompanied by different facial gestures, e.g. emphasis can be defined by a higher level of articulation, slightly lower pronunciation rate, and by raising eyebrows. Negation can be further emphasized by repetitive nods.

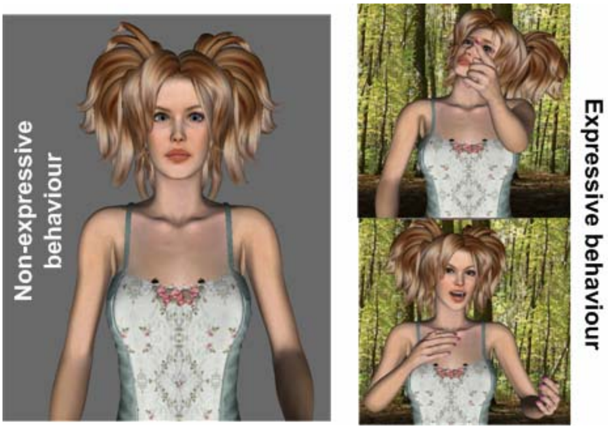


Fig. 11. Personification of the PLATTOS TTS system output.

The gestures used on the right-hand side (Expressive behaviour) of Figure 11 are independent and don’t directly influence each other. The animation blending technique enables the deployment of facial expressions, emotions, and speech, simultaneously. The bone-based ECA automatically removes most of the “jerky”, or unnatural poses that usually result when animating expressive ECAs, such as: the eyes don’t follow whilst the head is turning etc. Since the multipart concept uses a shared skeleton, even though the eyes and head are of different body types, the eyes will automatically be sub-parented to the joint chain of the head (to the one among the joints in the head joint-chain). This will result in the eyes following the head’s movements. Therefore, when the eyes move, head will be uninfluenced, but when the head moves, eyes will move according to an automatic gaze generation process. Furthermore, gestures, emotions, gaze and verbal communication (lip-sync), can vary in composition (which combinations of control points are used to form them), in amplitude (to what extent a gesture forms; e.g. co-articulation of utterances), in speed, and in repetitiveness. The expressive behaviour presented was generated by specifying each of the gesture types in the form of the EVA Script, provided by the PLATTOS TTS system. ECAs generated by the EVA framework and PLATTOS TTS system can generate different speech-driven types of gestures, gaze, and both simple and complex emotions, in an expressive, fully adjustable way. All the ECA's body movements within are defined and described hierarchically, as a composition of movements of the control units. The PLATTOS ECA-EVA enables the animation of rich sets of gestures, expressions, or event speech utterances that can vary in time, space and composition.

6. Evaluating multilingual and multimodal TTS systems

Constant evaluation as a constituent part of research activities has proven to be a successful approach for enhancing progress in almost all areas of speech technology, such as speech recognition, speech synthesis, or speech translation, especially if organized in the form of evaluation campaigns, e.g. TC-STAR[1], Blizzard[2] etc. (Rojc et al., 2009). As we know, the traditional evaluations are not performed 'on-line', the transport of test data and results has to be treated manually, and the test data are not 'secret'. Furthermore, the connecting of different developers' modules cannot be handled without an exchange of software to be integrated locally. In order to solve all these issues of traditional evaluations for testing TTS systems, a RES (remote evaluation system) evaluation framework has been established over recent years for speech synthesis technology within the ECESS consortium[3].

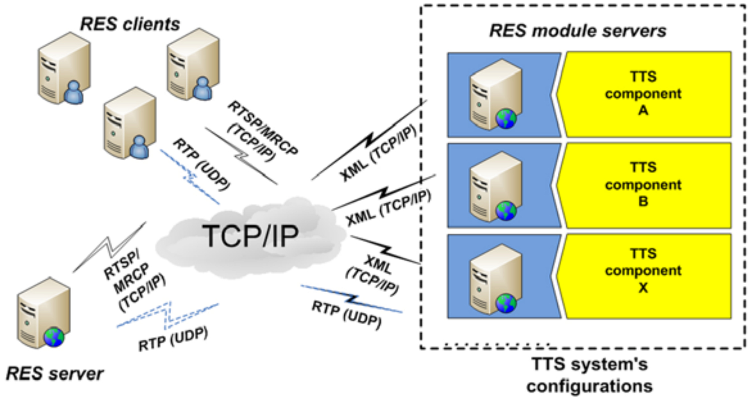


Fig. 12. RES framework for developing and evaluating multilingual and multimodal TTS systems and components.

The key element of the RES is its specification for a set of separate modules: e.g. for text processing, prosody generation, acoustic synthesis modules etc., that can be combined together into a complete text-to-speech system. Being able to split into any number of such modules has the advantage that the developers of an institution can concentrate its efforts on a single module, and test its performance within a complete system, using missing modules from the developers of other institutions etc. In this way high-performance multilingual and multimodal TTS systems can be built by using the high-performance modules of different institutions. A common evaluation methodology has been developed

[1] www.tc-star.org (EU project TC-STAR)
[2] http://festvox.org/blizzard/ (The Blizzard challenge)
[3] www.ecess.eu (ECESS - European Center of Excellence in Speech Synthesis)

to assess the performances of the modules that are based on the common use of those module-specific evaluation criteria and module-specific language resources needed for training and testing the modules. The RES was designed, not only to evaluate TTS modules, but also to support the developers of TTS modules. Developers/researchers can use RES in a test/development modus, in order to improve the performances of their TTS module(s), and evaluators can use RES in an evaluation modus for measuring the performances of the selected TTS modules. The distributed architecture of the RES is shown in Figure 12. As can be seen, the system consists of several RES clients (for developers, researchers, and evaluators), the RES server (managing unit), and RES module servers encapsulating the TTS modules (developers, and researchers). The RES server communicates simultaneously with several RES clients, and also supports the RES module servers when communicating with several RES clients at the same time. When performing testing or evaluating, developers, researchers and evaluators only select the desired TTS modules via RES clients and provide corresponding input for the selected task. The given input is then automatically transferred within the RES to the selected TTS modules, and generated output is returned to the RES client.

7. Conclusion

The presented design pattern for multilingual and multimodal corpus-based TTS systems shows that it is possible to integrate all modules of the TTS system, from text processing to acoustic processing, into an efficient and flexible queuing mechanism. Time and space-efficient FSMs are used for separating language dependent resources from a language-independent TTS engine, for the time and space-efficient representation of language resources, and for fast information lookup. A HRG structure is used for storing complex and heterogeneous sentence information, and for flexible construction of complex features. Furthermore, optimisation of the unit-selection process is one of the most important issues for corpus-based TTS systems, where several processing steps have important impacts on the achieved performance of the TTS system, regarding quality and efficiency. A RGD algorithm for cost functions' weights optimisation is proposed within the unit-selection process. Objective and subjective measures show that such optimisation results in a better quality of generated speech, a smaller common error 'E' regarding unit duration deviations and pitch disagreements between selected speech segments, is fully automatic and language-independent. It is, therefore, very helpful for tuning a general unit selection process, and can speed-up the generation of new voices for corpus-based TTS systems. The presented design pattern was demonstrated on the implementation of the Slovenian corpus-based PLATTOS TTS system; however, it can be used for the construction of TTS systems for other languages, for which the necessary language resources exist. By personification of the PLATTOS TTS system using ECA EVA, it can be used in advanced multimodal spoken dialogue systems. PLATTOS TTS system and EVA framework together provide flexible and efficient audio-visual multimodal output, enriched with a rich set of gestures, expressions, and emotions. Namely, by using EVA Script schemes, synthesized speech can be enhanced with several body movements, several types of visually represented articulation, different facial expressions (e.g. eye-lid movement, gaze, smile, emotions, etc.), and different body gestures (hand gestures, head movement, etc.). All these features personificate machine generated responses, and provide means for more natural human-machine interaction to be used in multimodal spoken dialogue systems. The ability, not only to articulate but also to control the speed and level of articulation, additionally enhances human-machine interfaces.

8. Acknowledgment

Operation part financed by the European Union, European Social Fund.

9. References

Ball, G. & Breese, J. (2000). Embodied conversational agents, Emotion and personality in a conversational agent, 2000

Black, A. W. & Taylor, P. (1997). Automatically clustering similar units for units selection in speech synthesis. Proceedings of Eurospeech 2, 601– 1368

Brill, E. (1993). A Corpus-Based Approach to Language Learning. Ph.D. Thesis

Campbell, N. & Black, A. (1996). CHATR: a multi-lingual speech resequencing synthesis system. Institute of Electronic, Information and Communication Engineers, Spring Meeting, Tokyo SP-96-07

Cassell, J. (2000). Nudge Nudge Wink Wink: Elements of Face-to-Face Conversation for Embodied Conversational Agents, in Cassell, J. et al. (eds.), *Embodied Conversational Agents,* pp. 1-27. Cambridge, MA: MIT Press

Daciuk, J. (1998). Incremental Construction of Finite-State Automata and Transducers and their Use in the Natural Language Processing, Ph.D. thesis, Technical University of Gdansk, Poland

DeCarolis B.; Pelachaud C.; Poggi I. & Steedman M. (2004). APML, a mark-up language for believable behavior generation. *In H. Prendinger and M. Ishizuka, editors, Life-like Characters. Tools, Affective Functions and Applications*, pp. 65-85, Springer-Verlag Berlin Heidelberg

Emmanuel, R. & Schabes, Y. (1995). Deterministic Part-Of-Speech Tagging with Finite-State Transducers. Mitsubishi Electric Research Laboratories

Georgantas, G.; Issarny, V. & Cerisara, C. (2006). Ambient Intelligence, Wireless Networking, and Ubiquitous Computing, chapter Dynamic Synthesis of Natural Human-Machine Interfaces in Ambient Intelligence Environments. Artech House, July 2006

Goslin, M. & Mine, M. R. (2004). The Panda3D Graphics Engine, Computer, v.37, n.10, p.112-114, October 2004

Gratch, J.; Rickel, J.; Andre, E.; Badler, N.; Cassell, J. & Petajan, E. (2002). Creating Interactive Virtual Humans: Some Assembly Required, *IEEE Intelligent Systems* 17(4): 54-6

Güdükbay, U.; Özgüç, B.; Memişoglu A. & Yeşil M.Š. (2008). Modeling, Animation, and Rendering of Human Figures. Signals and Communication Technology, 2008, pages 201-238, Springer

Heloir & Kipp, M. (2009). EMBR—a realtime animation engine for interactive embodied agents. *In Proceedings of the 9th International Conference on Intelligent Virtual Agents (IVA '09)*, pp. 393–404, Springer, Amsterdam, The Netherlands, 2009

Holzapfel, M. & Campbell, N. (1998). A nonlinear unit selection strategy for concatenative speech synthesis based on syllable level features, In ICSLP-1998, paper 0521, Sydney, Australia

Holzapfel, M. (2000). Konkatenative Sprachsynthese mit grossen Datenbanken. Ph.D. Thesis.

Mlakar, I. & Rojc, M. (2011). EVA: expressive multipart virtual agent performing gestures and emotions. International journal of mathematics and computers in simulation, 2011, vol. 5, iss. 1, pp. 36-44

Mohri, M. (1995). On Some Applications of Finite-State Automata Theory to Natural Language Processing. Natural Language Engineering, 1

Mohri, M.; Fernando, C. N. Pereira & Riley, M. (1996). Weighted automata in text and speech processing. In: ECAI-96 Workshop, Budapest, Hungary. ECAI

Mohri, M. & Sproat, R. (1996). An efficient compiler for weighted rewrite rules. In: 34-th Meeting of the Association for Computational Linguistics (ACL 96), Santa Cruz, California

Mohri, M. (1997). Finite-state transducers in language and speech processing. Computational Linguistics 23 (2), 269–311

Poggi I. & Pelachaud C. (2000). Performative facial expressions in Animated faces, Conversational Agents, Emotion and personality in a conversational agent, In Book Embodied conversational agents, MA, USA, 2000

Raitio, T.; Suni, A.; Yamagishi, J.; Pulakka, H.; Nurminen, J.; Vainio, M. & Alku, P. (2011). HMM-based speech synthesis utilizing glottal inverse filtering. IEEE Transactions on Audio, Speech and Language Processing, 19(1):153-165, January 2011

Rojc, M. & Kačič, Z. (2000). A computational platform for development of morphologic and phonetic lexica. In Proceedings of the Second Language Resources and Evaluation Conference (LREC), Athens, Greece

Rojc, M. (2000). The Use of Finite-state Machines for Text-to-Speech Systems. Master thesis

Rojc, M. (2003). Time and Space Efficient Architecture of the Multilingual and Polyglot Text-to-Speech System – Architecture with Finite-State Machines. Ph.D. Thesis

Rojc, M. & Kačič, Z. (2007). Time and Space-Efficient Architecture for a Corpus-based Text-to-Speech Synthesis System, Speech Communication, Vol. 49 (3), 2007, pp. 230-249

Rojc, M. & Kačič, Z. (June 2007). A Unified Approach to Grapheme-to-Phoneme Conversion for the PLATTOS Slovenian Text-to-Speech System, Applied Artificial Intelligence, Vol. 21 (6), June, 2007, pp. 563-603

Sproat, R.; Riley, M. (1996). Compilation of weighted finite-state transducers from decision trees. In: Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics, pp. 215–222

Sproat, R. (1998). Multilingual Text-to-Speech Synthesis. Kluwer Academic Publishers, ISBN 0-7923-8027-4, Massachusetts, USA

Taylor, P.; Black, A. & Caley, R. (1998). The architecture of the Festival speech synthesis system. In: Proceedings of the Third ESCA Workshop in Speech Synthesis, pp. 147–151

Taylor, P.; Black, A. & Caley, R. (2001). Heterogeneous relation graphs as a formalism for representing linguistic information. Speech Communication 33 (1-2), 153–174

Vilhjalmsson, H.; Cantelmo, N.; Cassell, J.; Chafai, N.; Kipp, M.; Kopp, S.; Mancini, M.; Marsella, S.; Marshall A.; Pelachaud, C.; Ruttkay Z.; Thorisson, K.; van Welbergen, H. & van der Werf, R. (2007). The Behavior Markup Language: Recent Developments and Challenges. IVA 2007, LNAI 4722: 99-111, Springer-Verlag Berlin Heidelberg

8

Estimation of Speech Intelligibility Using Perceptual Speech Quality Scores

Kazuhiro Kondo
Graduate School of Science and Engineering, Yamagata University
Japan

1. Introduction

Recent advances in mobile wireless communication devices have made possible speech communication in a variety of noise environments which were not possible before. Also, sophisticated speech encoders, echo control devices, and noise canceling devices have caused artificial synthetic noise, *e.g.* musical noise, which were not seen before with analog or simple PCM speech communication. Thus, a need for comprehensive speech communication quality measures and frequent evaluation efforts have become a necessity. Speech quality is generally measured in one of two measures. The overall listening quality, such as the "naturalness" of the test speech, is typically measured as the Mean Opinion Score (MOS) (ITU-T, 1996). The other criteria is speech intelligibility, which tries to measure the accuracy with which the test speech material carries its spoken content. We will deal mainly with the latter measure in this chapter.

There were not many variations in the types of degradations seen in conventional speech communication systems. Common types of degradations seen were simple ones, such as band limitation and additive noise. Thus, evaluation procedures were fairly simple. Traditionally, Japanese intelligibility tests often used stimuli of randomly selected single mora, two morae or three morae speech (Iida, 1987). The subjects were free to choose from any combination of valid Japanese syllables. This quickly became a strenuous task as the channel distortion increases. Thus, intelligibility tests of this kind is known to be unstable and often do not reflect the physically evident distortion, giving surprising results (Nishimura et al., 1996).

English intelligibility tests are also reported to show similar trends. Accordingly, the Diagnostic Rhyme Test (DRT) (Voiers, 1977; 1983), a closed set selection test that restricted the reply to two words, was proposed. This test is said to be effective in controlling various factors including the amount of training and phonetic context, and is known to give stable intelligibility scores. The DRT has now become an ANSI standard (ANSI, 1989).

In this chapter, we will briefly describe a DRT-type closed set selection test in Japanese (Kondo et al., 2007; 2001). We categorized Japanese consonants into the same taxonomy used for the English tests, and proposed a minimum-pair list accordingly which differ only by the initial consonant and by a single phonetic feature. Subjective test results are also shown with various noise under various SNR.

Then, we will investigate on methods to estimate intelligibility through objective measures. If this is possible with reasonable accuracy, we should be able to "screen" the intelligibility in many of the conditions, and limit the need for full-scale subjective test to a minimum subset.

Feature	m	n	z	J̌	b	d	g	w	r	j	ɸ	s	š	č	p	t	k	h	N	ts	ç
Voicing (vocalic-nonvocalic)	+	+	+	+	+	+	+	+	+	+	–	–	–	–	–	–	–	–	+	–	–
Nasality (nasal-oral)	+	+	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	+	–	–
Sustention (continuant-interrupted)	–	–	+	–	–	–	–	+	+	+	+	+	+	–	–	–	–	+	–	–	+
Sibilation (strident-mellow)	–	–	+	+	–	–	–	–	–	–	–	+	+	+	–	–	–	–	–	+	–
Graveness (grave-acute)	+	–	–	0	+	–	0	+	–	0	+	–	0	0	+	–	0	0	0	–	0
Compactness (compact-diffuse)	–	–	–	+	–	–	+	–	–	+	–	–	+	+	–	–	+	+	–	–	+
Vowel-like (glide-nonglide)	–	–	–	–	–	–	–	+	+	+	–	–	–	–	–	–	–	–	–	–	–

Table 1. The Japanese consonant taxonomy

We will describe our efforts using PESQ (Perceptual Evaluation of Subjective Quality) scores, an ITU standard which estimates MOS from both degraded and original speech, and try to map PESQ-derived MOS to intelligibility.

2. The Japanese diagnostic rhyme test

2.1 Diagnostic rhyme test

Diagnostic Rhyme Tests (DRT) are speech intelligibility tests that forces the tester to choose one word that they perceived from a list of two rhyming words. The two rhyming words differ by only the initial consonant by a single distinctive feature.

DRT assumes the following simplification and principles which will enable even naive listeners to provide stable and efficient intelligibility scores (Voiers, 1977; 1983).

- Additive and convolutional noise mostly affect consonants, which carry the bulk of linguistic information, and not vowels. Thus, exact reproduction of consonants are essential in voice communications. This is also the basis for the Fairbanks Rhyme Tests (Fairbanks, 1958), which tested only consonant recognizability.
- Consonant apprehensibility in the initial, intervocalic and final positions are strongly correlated. Thus, one can measure apprehensibility in all positions just by measuring at the initial position. This assumption is backed by experiments by Suzuki *et al.* (Suzuki et al., 1998), in which they found that there is a strong correlation in the articulation scores of the first and second mora.
- The effect of word familiarity and phonetic context can be neglected if the number of response choices (Miller et al., 1951; Voiers, 1977). In the case of the DRT, the response is restricted to one word out of a pair of words.

In accordance with these assumptions, the DRT uses word-pairs which are minimal pairs in which only the initial consonant differs by a single phonetic attribute as defined by Jakobson, Fant, and Halle (Jakobson et al., 1952). The choice of word-pairs from which the listener selects their response always contains the correct word.

2.2 The Japanese consonant taxonomy

We first proposed a consonant taxonomy for Japanese with the same feature classification used in English, which were drawn from the classification by Jakobson, Fant and Halle (Jakobson et al., 1952) (to be denoted as JFH classification). Table 1 shows the proposed Japanese consonant taxonomy. The "+" shows that the feature is present, the "-" shows the absence, and “0” shows that the feature does not apply to the consonant. The following seven features were used.

1. Voicing: corresponds to the vocalic-nonvocalic classification by JFH. This is a trivial classification.
2. Nasality: corresponds to the nasal-oral classification by JFH. This is also a fairly trivial classification.
3. Sustention: corresponds to the continuant-interrupted classification. This classifies consonants into clearly continuous consonants and other transient phones, such as plosives.
4. Sibilation: corresponds to the strident-mellow classification. This roughly corresponds to the randomness of the consonants.
5. Graveness: corresponds to the grave-acute opposition. If the spectrum of the consonant concentrates in the low frequency region, it is classified as grave, and vice versa. Also, the oral cavity is not obstructed with grave consonants, while with acute consonants, the oral cavity is divided into compartments with the tongue.
6. Compactness: corresponds to the compact-diffuse opposition. If the spectrum of the consonant largely concentrates around the formant, it is classified as compact, and vice versa.
7. Vowel-like: this classification is not used. It classifies consonants into glides and other true consonants.

We classified most consonants in Japanese speech roughly in the same manner as English. However, several exceptions were noted.

- The consonant [g] is often nasalized in intervocalic positions. However, since we are only dealing with initial consonants, this consonant was classified as oral. Thus, nasality was classified as “-” (feature absent).
- Allophones such as [ŋ] were not classified.

2.3 The Japanese DRT word-pair list

The consonant taxonomy was then used to compile a word-pair list to be used as stimuli for the DRT. Ten word-pairs per each of the 6 features, one pair per each of the five vowel context, were proposed for a total of 120 words (Fujimori et al., 2006; Kondo et al., 2007). The word-pairs are rhyme words, differing only in the initial phoneme. The proposed word-pair list is shown in Table 2. The first words in the word-pair list are words whose initial consonants have the consonant feature under test, and the initial consonants in the latter words do not. Note that all five vowel context are covered.
The following is specific for the Japanese list:

- Only two morae words were initially considered. Longer words will be considered as needed.
- Foreign words were avoided when possible. However, words starting with the [p] context are mostly foreign words, and thus foreign words were included in this case.
- Only words with the same accent type were selected as a word-pair.
- We tried to select mostly common nouns. Proper nouns, slang words and obscure words were avoided where possible.

Voicing	Nasality	Sustention	Sibilation	Graveness	Compactness
Zai - Sai	Man - Ban	Hashi - Kashi	Jamu - Gamu	Waku - Raku	Yaku - Waku
Daku - Taku	Nai - Dai	Hata - Kata	Chaku - Kaku	Pai - Tai	Kai - Pai
Giji - Kiji	Misu - Bisu	Shiri - Chiri	Shiki - Hiki	Mie - Nie	Gin - Bin
Gin - Kin	Miru - Biru	Hiru - Kiru	Chiji - Kiji	Misu - Nisu	Kiza - Piza
Zui - Sui	Muri - Buri	Suki - Tsuki	Chuu - Kuu	Muku - Nuku	Kuro - Puro
Guu - Kuu	Mushi - Bushi	Suna - Tsuna	Jun - Gun	Mushi - Nushi	Yuu - Ruu
Zei - Sei	Men - Ben	Hen - Ken	Shea - Hea	Men - Nen	Gen - Ben
Deba - Teba	Neru - Deru	Heri - Keri	Sheru - Heru	Pen - Ten	Ken - Pen
Zoo - Soo	Mon - Bon	Hoshi - Koshi	Joo - Goo	Moo - Noo	Goki - Boki
Goji - Koji	Nora - Dora	Horu - Koru	Shoji - Hoji	Poro - Toro	Yoka - Roka

Table 2. Japanese DRT word-pair list

- Words which include double consonants and palatalized syllables were excluded when possible.

Additionally, rare consonant-vowel combinations were substituted with other syllables where possible.
As stated before, familiarity may affect the intelligibility scores, although using word-pairs will most likely mitigate this effect. However, to be safe, we selected words which have relatively high phonetic-text familiarity (average 5.5, standard deviation 0.72 on a 7-point scale) according to the familiarity listing compiled by Amano *et al.* (Amano & Kondo, 1999).
Word accents types were judged with reference to both (Amano & Kondo, 1999) and (NHK Broadcasting Culture Research Institute, 1998). Over 77 % of the words were accent type 1 (high to low pitch accent transition), and 2 % were type 0 (flat). Both words in the word-pair had the same accent type. When multiple accent types exist, the speakers were asked to record using the specified accent type, with the same accent type as the other word in the word-pair. The recorded speech was checked for clear pronunciation and accent, and re-recorded as needed.

2.4 The DRT evaluation procedure

Words spoken by multiple speakers should be used. At least 8 listeners should be employed for the test. The listener listens to the stimulus word speech, and selects the correct answer from one of the words in the word-pair. The ordering of the stimulus can be completely random, or it can cycle through the vowel context (*i.e.* form a 5-word cycle covering the five vowel context). The intelligibility is measured by the average correct response rate over each of the six consonant features, or by the average over all features. The correct response rate (CACR) should be calculated using the following formula to compensate for the chance level,

$$S = \frac{100(R - W)}{T} [\%] \tag{1}$$

where S is the response rate adjusted for chance ("true" correct response rate), R is the observed number of correct responses, W the observed number of incorrect responses, and T the total number of responses. In other words, since this is a two-to-one selection test, a completely random response will result in half of the responses to be correct. With the above formula, completely random response will give average response rate of 0 %.

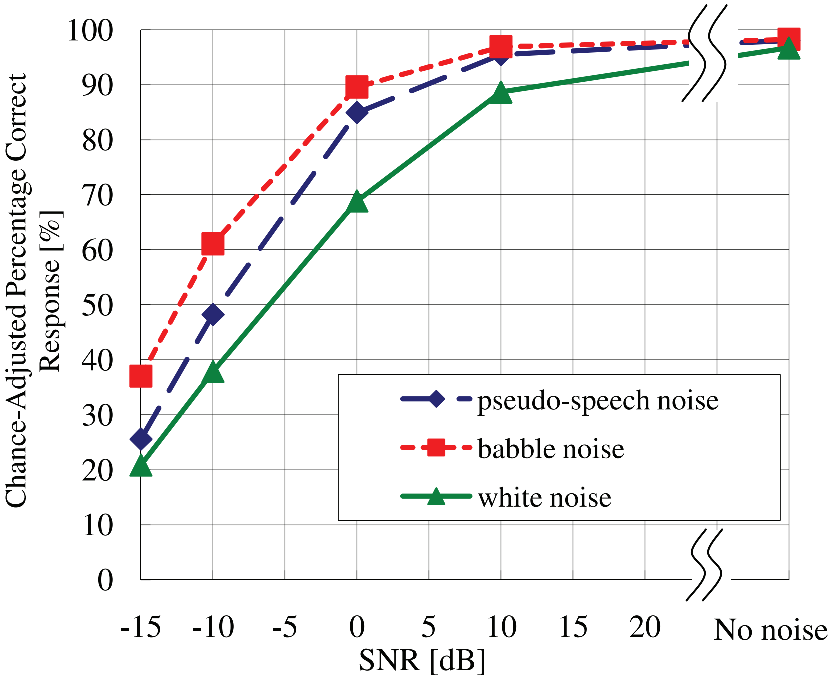


Fig. 1. Comparison of DRT scores for speech with three types of noises

2.5 DRT evaluation experimental setup

We evaluated the DRT on a relatively large Japanese speech database with three typical noise types in order to compare its sensitivity to noise with DRT results in English. We collected speech from eight untrained speakers, four male (all in their twenties) and four female (three in their twenties, and one in her fifties). All words in the DRT word list were recorded using a head-mount electret microphone (Sennheiser HD 410-6) at a sampling rate of 16 kHz, 16 bits per sample. No directions on the pronunciation and accents were initially given, so that the speakers would be able to speak naturally. Re-recordings were made as needed when the speech samples were not of standard accent, or unclear. White noise, multi-speaker (babble) (Rice University, 1995) and pseudo-speech noise (Tanaka, 1989) were mixed into these samples at an SNR of -15, -10, 0 and 10 dB, respectively. Speech for words in the word-pair list was played out in random order. All speech were played out diotically through headphones (Sennheiser HD 25-1 II) at the listener's preferred output level. The listeners were shown both words in the word-pair to choose from. Eleven listeners underwent the tests for speech mixed with white noise, and 5 listeners tested speech in pseudo-speech and babble noise. All listeners were native Japanese speakers in their twenties with reportedly normal hearing. Each listener listened to 8 speakers, 5 noise levels including clean, 6 phonetic features, and 20 words per feature, bringing the total to 4800 spoken words per noise type.

2.6 Results and discussion

Figure 1 shows the average DRT scores (the chance adjusted correct response percentage, CACR) over all phonetic features for the three types of noise tested. Figures 2, 3, and 4 show

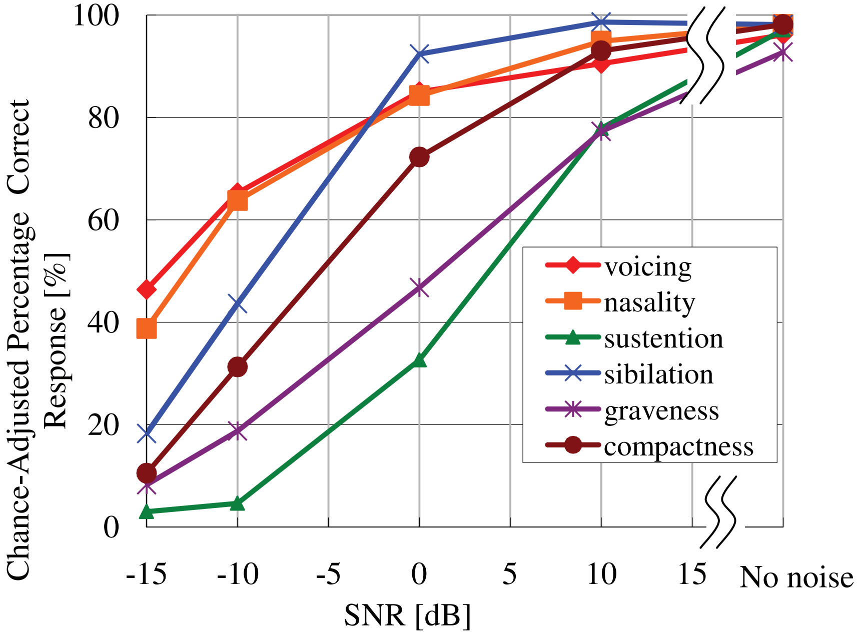


Fig. 2. DRT scores for speech mixed with white noise

CACRs for each of the mixed noise types by the phonetic feature. Two-way ANOVA tests have confirmed SNR and phonetic feature to be main effects in all noise types tested.
The overall trend for all noise types generally agrees with English results obtained by Voiers (Voiers, 1977; 1983). The following can be drawn from the results:

1. The average DRT score over all phonetic feature vs. SNR is similar regardless of the noise type. However, white noise seems to affect the scores most, followed by pseudo-speech noise, and babble. The reason for this seems to be the bandwidth of the noise, especially in the high frequency regions.
2. Sibilation generally shows high scores when white noise level is low. However, the scores decrease quickly as white noise level increases. This again agrees well with results by Voiers (Voiers, 1977; 1983). The reason for this can be that phones with sibilation show wide frequency bandwidth, similar to white noise. This may also be the reason the scores are not affected as much by other types of noise since these have much narrower bandwidth.
3. Much less difference by features is seen with pseudo-speech and babble noise compared to white noise. In other words, each of the phonetic feature is affected similarly with these noise types. Nasality, sustention, and compactness especially show insignificant differences. This was observed in English tests as well. The reason for this again may be the bandwidth of the added noise.

Figure 5 compares the DRT scores for white noise-added speech by speaker gender. As shown by this figure, the DRT scores are virtually same for both male and female speech for all ranges of SNR tested, and thus the gender of the speaker has insignificant effect on the DRT scores. This was also confirmed with ANOVA.

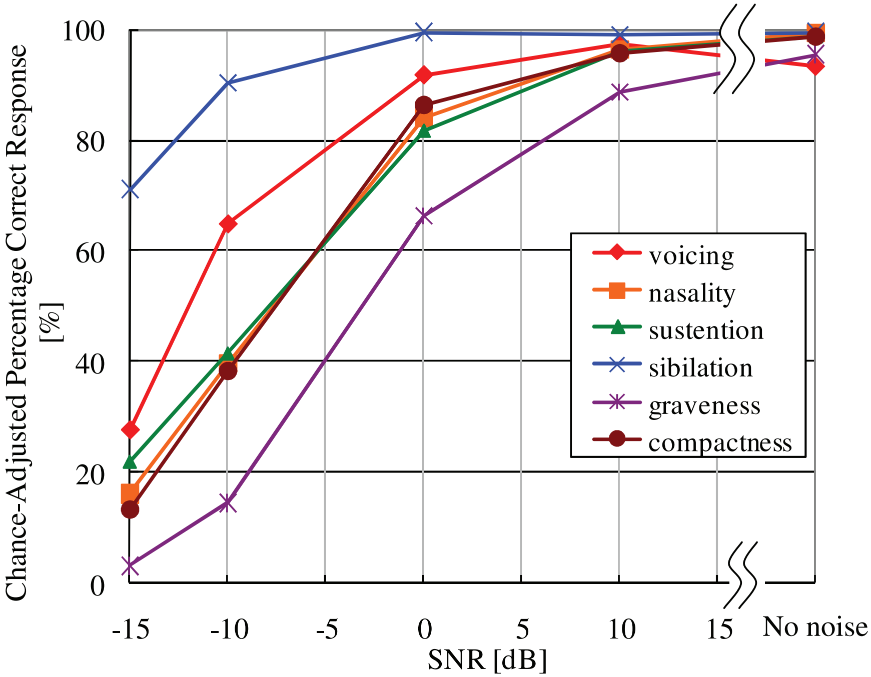


Fig. 3. DRT scores for speech mixed with pseudo-speech noise

3. Estimation of DRT scores using objective measures

In this section, we will describe our approach to estimating the subjective intelligibility DRT scores using objective measures. Even though the proposed DRT tests were much simpler than conventional intelligibility tests, the DRT test still requires human listeners to rate more than one hundred words per noise condition. Accordingly, in the following, we attempted to estimate subjective DRT scores using objective measures obtained by some calculations without human participants. If estimation of intelligibility, at least to some degree, is possible, we should be able to "screen" the intelligibility in many of the conditions, and limit the need for full-scale subjective tests to a minimum.

3.1 Estimation of DRT using PESQ

In this section, we will describe results of experiments to estimate DRT scores from PESQ (Perceptual Evaluation of Speech Quality) scores (Kaga et al., 2006). PESQ is an international standard which tries to estimate subjective Mean Opinion Scores (MOS) (ITU-T, 1996) from the original and the degraded signal (Beerends et al., 2002; ITU-T, 2001; Rix et al., 2002). PESQ is known to be one of the most accurate objective methods to estimate subjective MOS. Although MOS is a subjective measure of the overall speech quality, we can assume that speech quality is "loosely" correlated with speech intelligibility. Thus, we can assume that speech intelligibility is related to estimated MOS values, at least to some degree.

Kitawaki and Yamada have recently conducted a small scale test to employ PESQ to estimate word intelligibility (Kitawaki & Yamada, 2007). They used speech categorized into four classes of word familiarity. They found relatively high correlation between subjective word

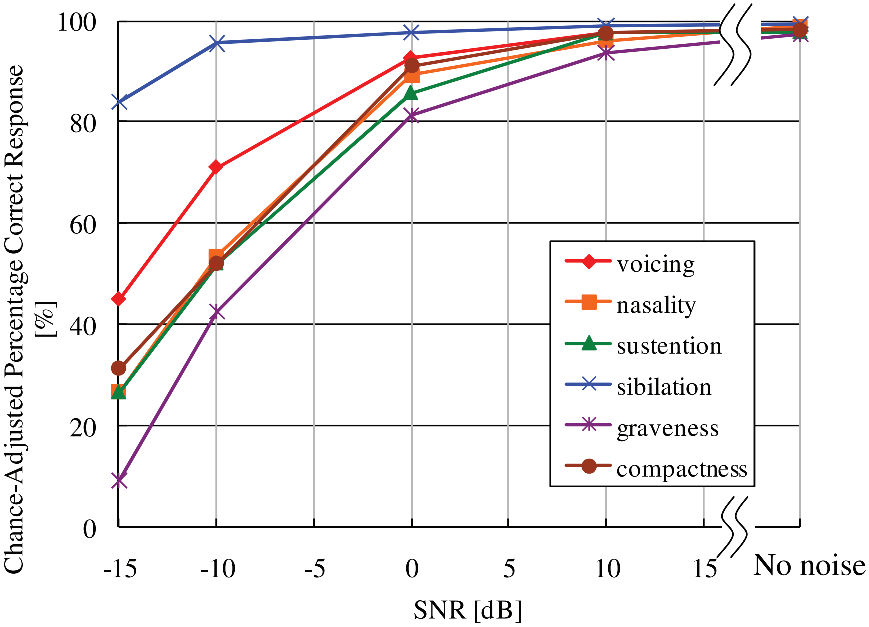


Fig. 4. DRT scores for speech mixed with babble noise

intelligibility and estimated word intelligibility using PESQ scores, especially when the word familiarity is low. Beerends *et al.* also used PESQ to estimate intelligibility (Beerends et al., 2009). They found that PESQ fails to predict intelligibility especially at lower SNR. Thus, they use several methods to improve the estimation in this region, *e.g*, the use of spectral subtraction, silent interval deletion, and steady-state suppression. They show some success in improving the accuracy. On the other hand, Liu *et al.* have recently attempted to estimate speech intelligibility from a number objective measures including PESQ scores (Liu et al., 2008). They used digits for their speech samples, and found very low correlation between intelligibility and PESQ scores. In fact, they found low correlation in most of the objective measures they attempted, highlighting the difficulty of this problem.

3.2 Perceptual Evaluation of Speech Quality (PESQ)

The Perceptual Evaluation of Speech Quality (PESQ) (ITU-T, 2001; 2003; 2005) is an international standard for estimating the Mean Opinion Score (MOS) from both the clean and degraded signal. It evolved from a number of prior attempts to estimate MOS, and is regarded as one of the most sophisticated and accurate estimation methods available today. PESQ was officially standardized by the International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) as recommendation P.862 in February, 2001, and extended to wideband speech as recommendation P.862.2 in November, 2005. A simplified diagram of the PESQ is shown in Fig. 6.

PESQ uses a perceptual model to covert the input and the degraded speech into an internal representation. The degraded speech is time-aligned with the original signal to compensate for the delay that may be associated with the degradation. The difference in the two internal representation is then used by the cognitive model to estimate the MOS.

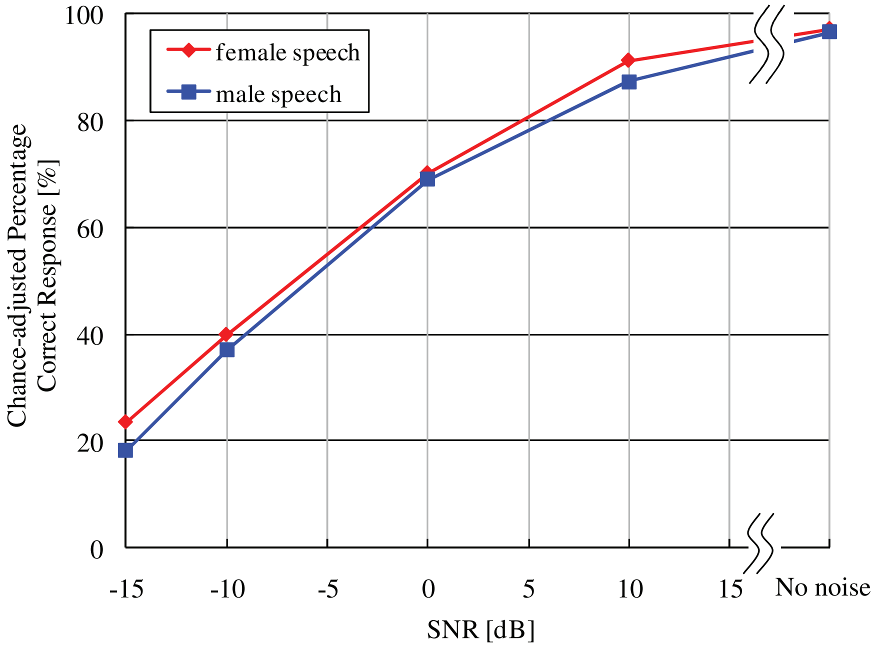


Fig. 5. Comparison of DRT scores of speech mixed with white noise by speaker gender

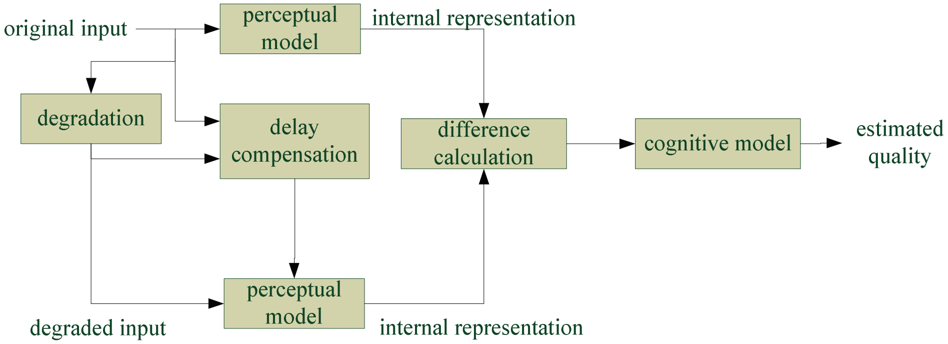


Fig. 6. Simplified diagram of the PESQ algorithm

Figure 7 is the result of an experiment we conducted to estimated the MOS-LQO (Listening Quality Objective), which is an estimated MOS output of the PESQ algorithm (ITU-T, 2003). We used read Japanese sentences of two male and two female speaker, five per speaker for a total of 20 sentences. White noise was added to these speech samples at 30, 10, and -5dB. We also encoded and decoded speech samples with the G.729 CS-ACELP codec (ITU-T, 2007). This codec is commonly used in VOIP applications nowadays. All samples were sampled at 8 kHz, 16 bits per sample. The MOS-LQO for all degraded samples were estimated using PESQ. We also ran MOS tests using 10 listeners with the same degraded samples and the original speech. As can be seen this figure, the estimated MOS-LQO generally agrees well with subjective MOS. The line included in the figure is the fitted line using least mean square

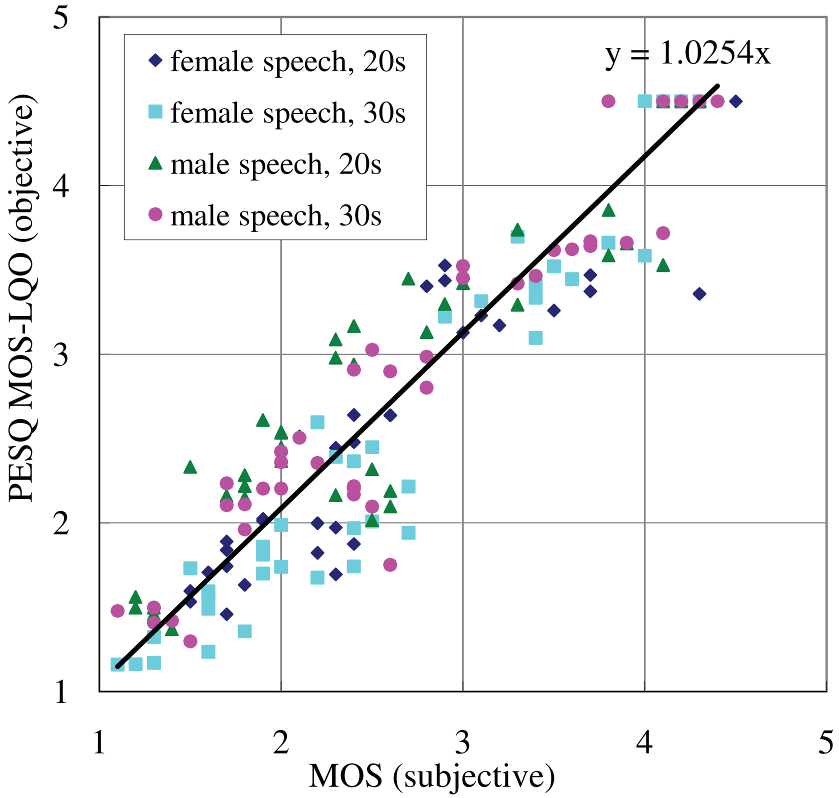


Fig. 7. Example MOS estimation using PESQ

error, which came out to be a gradient of 1.024, also showing that the estimated MOS-LQO generally are accurate estimation of the subjective MOS.

3.3 Correlation between PESQ MOS-LQO and DRT intelligibility score (CACR)

We selected two male and two female speech with standard Japanese DRT words from the collected data described in section 2. All samples were sampled at 16 kHz, 16 bits monaurally. These speech samples were mixed with white noise and babble (Rice University, 1995) at SNR of -15, -10, 0 and 10 dB. Standard subjective DRT tests were run with these samples. Ten listeners were employed. We also estimated MOS-LQO of the degraded samples using PESQ. The wide-band option (+wb) was used in all tests.

Figures 8, and 9 plots the estimated MOS-LQO using PESQ against the corresponding DRT Chance-Adjusted Correct Response (CACR) for speech mixed with white noise, and babble noise, respectively. The speech was pooled for both speakers, for all of the SNR tested in each Figure. As can be seen in both noise types, the correlation between raw MOS-LQO and CACR is quite low. In fact, the Pearson correlation coefficient is 0.47 and 0.44 for female and male speech mixed with white noise, and 0.36 and 0.42 for babble noise. Most of the MOS-LQO is close to the lower end of the scale, *i.e.* well below 2.0, close to 1.0. This is not surprising since PESQ was designed to estimate MOS, and not intelligibility. MOS generally measures the overall speech quality with relatively small degradation, *i.e.* high SNR range, typically well above 0 dB. However, as we have seen in the previous section, intelligibility is measured in the lower SNR range, typically -20 to 0 dB. Thus we need to re-map the MOS-LQO to match the SNR range of interest for intelligibility estimation.

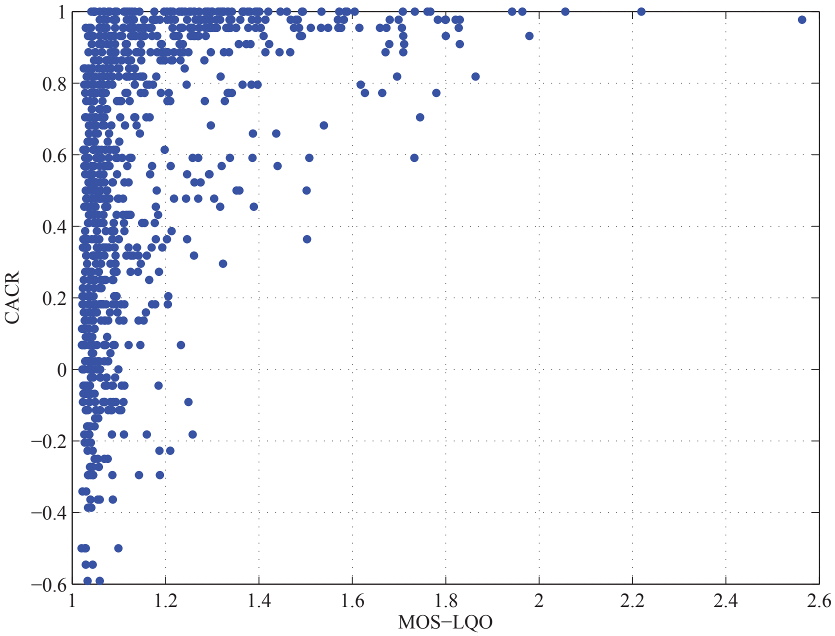


Fig. 8. MOS estimation of DRT words using PESQ (white noise)

3.4 Estimation of intelligibility by mapping per-word MOS-LQO to DRT CACR

We now attempt to map MOS-LQO to CACR using polynomial mapping. We estimated a quadratic polynomial to map the estimated MOS-LQO to DRT CACR on one training speaker. Then we used this polynomial to map MOS-LQO of a different test speaker to DRT CACR. The mapping was estimated for each noise type since it is reasonable to assume that we can obtain a small sample of the noise environment in which we want to estimate the DRT CACR beforehand. We also estimate one polynomial for each phonetic feature, as well as over all features. Table 3 shows an example of the estimated coefficients of the polynomials used to map the female speech mixed with white noise, $y = a_1x^2 + a_2x + a_3$, where x is the MOS-LQO, and y the estimated CACR. As can be seen, the coefficients differ significantly by phonetic feature. The coefficients were also shown to differ significantly by noise type or speaker gender as well.

Tables 4 through 7 tabulate the root mean square DRT CACR estimation error, and the Pearson correlation between subjective and estimated DRT CACR for speech (female and male) mixed with white noise and babble noise, respectively. As can be seen, average estimation errors range from approximately 0.2 to close to 0.7 in some cases. The correlation also ranges from 0.7 to virtually 0.0 in one extreme cases. Thus, the estimation accuracy varies widely by the phonetic feature. Estimation over all features generally perform worse than when using a single phonetic feature.

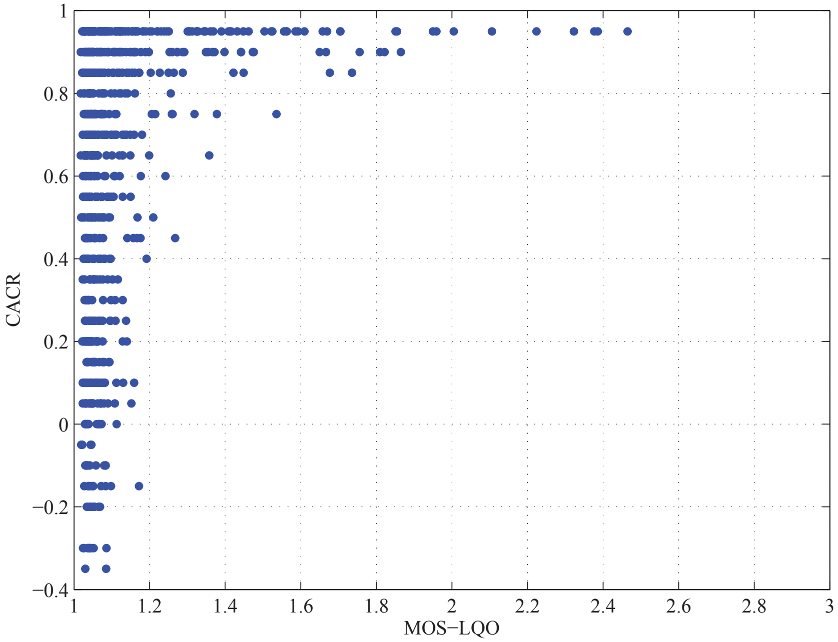


Fig. 9. MOS estimation of DRT words using PESQ (babble noise)

Phonetic feature	a_1	a_2	a_3
voicing	-0.853	2.80	-1.35
nasality	-1.93	6.15	-3.75
sustention	-3.28	10.3	-7.06
sibilation	-1.50	5.15	-3.26
graveness	-1.75	6.02	-4.22
compactness	0.76	-0.893	0.513
all features	-1.84	6.06	-3.94

Table 3. Polynomial coefficients of the mapping function used to map PESQ MOS-LQO to DRT CACR (white noise, female speech)

Figure 10 plots the subjective DRT CACR vs. the estimated DRT CACR for female speech samples for sustention mixed with white noise. This is one of the combinations showing the lowest RMSE and the highest correlation, *i.e.* one of the best predictions. However, the plots scatter widely from the equal rate line. Still the plots are evenly spaced around the equal rate line, and the best fit line is almost equal to the equal rate line. This gives us a clue leading to the approach taken in the next section.

3.5 Estimation of intelligibility by mapping per-feature MOS-LQO to DRT CACR

The standard procedure to measure the subjective intelligibility of a phonetic feature, as measured by CACR, is to test all 20 words on a large listener population, and average

Phonetic feature	RMSE	Correlation
voicing	0.20	0.51
nasality	0.23	0.59
sustention	0.26	0.77
sibilation	0.34	0.54
graveness	0.30	0.63
compactness	0.34	0.49
all features	0.65	0.23

Table 4. Root mean square estimation error and correlation of DRT CACR estimated from PESQ MOS-LQO (white noise, female speech)

Phonetic feature	RMSE	Correlation
voicing	0.68	-0.06
nasality	0.21	0.65
sustention	0.30	0.67
sibilation	0.35	0.52
graveness	0.32	0.61
compactness	0.39	0.41
all features	0.33	0.55

Table 5. Root mean square estimation error and correlation of DRT CACR estimated from PESQ MOS-LQO (white noise, male speech)

Phonetic feature	RMSE	Correlation
voicing	0.27	0.42
nasality	0.33	0.50
sustention	0.32	0.50
sibilation	0.08	0.38
graveness	0.29	0.66
compactness	0.28	0.54
all features	0.52	0.26

Table 6. Root mean square estimation error and correlation of DRT CACR estimated from PESQ MOS-LQO (babble noise, female speech)

the correct response rates for each of the conditions, *e.g.* noise type, SNR, etc. This is because subjective test inherently include a large degree of variations, both due to the tester individuality, and due to variations in the acoustics of the test word speech. By averaging the results for sufficiently large population of testers and over all words in the test list, we can expect to obtain stable reproducible results.

We will attempt the same procedure used to calculate the subjective CACR with the estimated per-word CACR to obtain the per phonetic feature DRT CACR. We pooled all CACR for a

Phonetic feature	RMSE	Correlation
voicing	0.49	0.16
nasality	0.31	0.58
sustention	0.30	0.54
sibilation	0.08	0.36
graveness	0.31	0.61
compactness	0.30	0.56
all features	0.31	0.30

Table 7. Root mean square estimation error and correlation of DRT CACR estimated from PESQ MOS-LQO (babble noise, male speech)

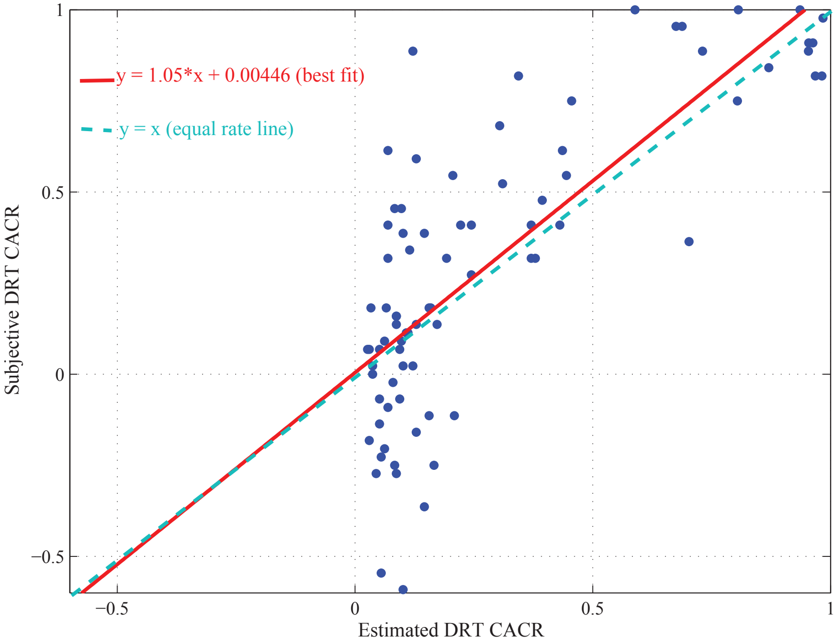


Fig. 10. Subjective CACR vs. estimated CACR (sustention, female speech with white noise)

single phonetic feature, per noise level (SNR) and type, into one CACR. The same quadratic polynomial mapping is used to map the MOS-LQO to DRT CACR, one mapping function per phonetic feature. Again, the mapping was calculated on one training speaker, and this mapping function was used to map MOS-LQO obtained using the PESQ algorithm to calculate the estimated DRT CACR for a different test speaker.

Figures 11 and 12 plot the subjective DRT CACR vs. estimated DRT CACR by pooling for female and male speech in white noise, respectively, while Figures 13 and 14 plot the DRT CACR for female and male speech in babble noise, respectively. Compared to Fig. 10, all plots in these figures are generally much closer to the equal rate line, as expected. This is the result of averaging out the deviation that was present with each of the words in the test word per phonetic feature. However, as can be seen in Fig. 14, we do not see any estimated DRT CACR below 0.4 for male speech in babble noise. This is due to the limited range that is seen with MOS-LQO under these conditions.

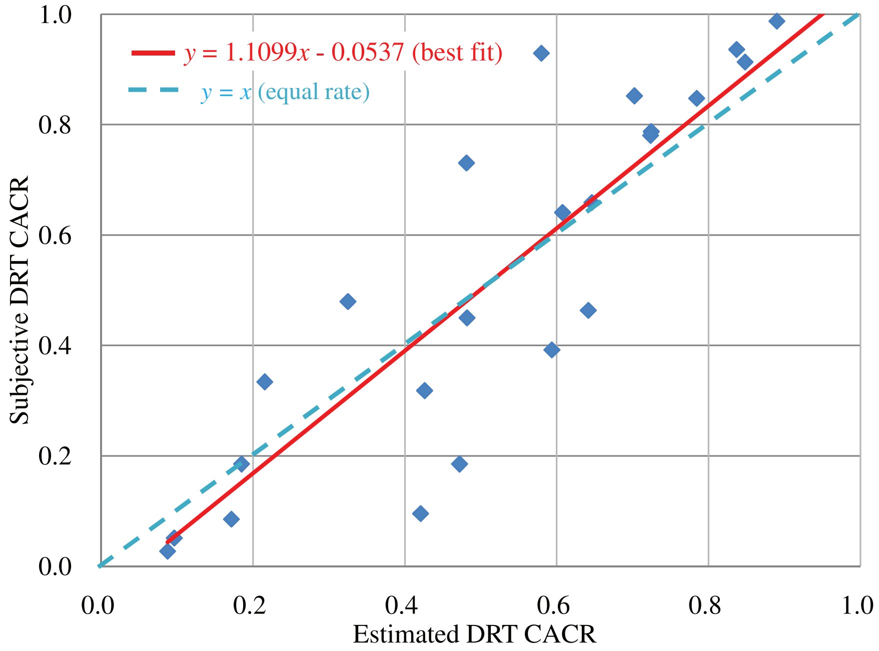


Fig. 11. Subjective CACR vs. estimated CACR (pooled for each feature, female speech with white noise)

Table 8 tabulates the root mean square estimation error and the correlation between subjective and estimated DRT CACR. The RMSE decreased to below 0.2, but even more surprising is the correlation, which is generally above 0.8 now. This level of accuracy is well within practical range if we want to "screen" tested conditions before testing with actual human listeners, as was stated as the goal of this research.

Noise	speaker gender	RMSE	Correlation
white	female	0.15	0.88
white	male	0.20	0.80
babble	female	0.18	0.78
babble	male	0.17	0.82

Table 8. Root mean square estimation error and correlation of DRT CACR estimated from pooled PESQ MOS-LQO

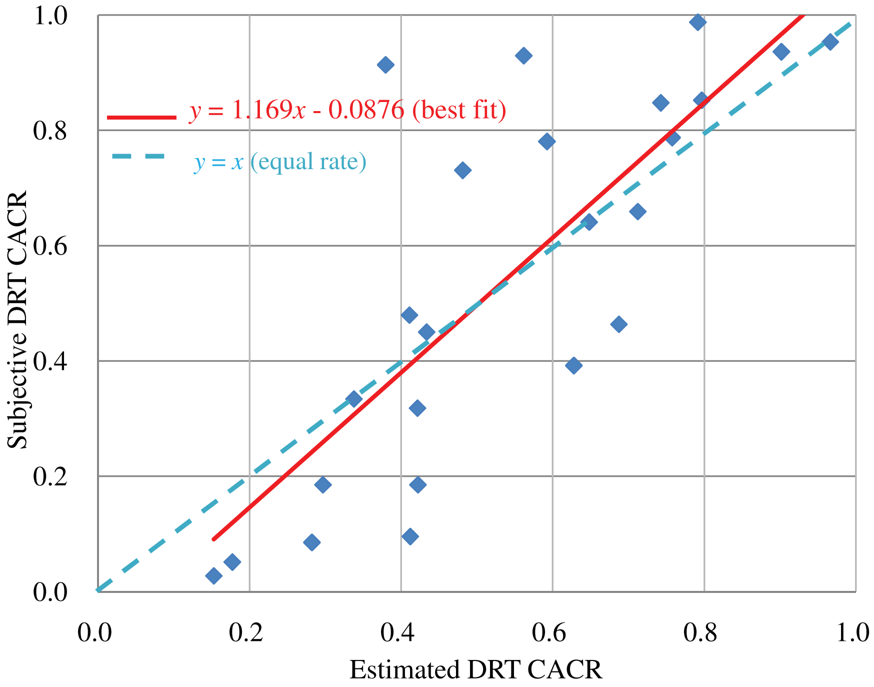


Fig. 12. Subjective CACR vs. estimated CACR (pooled for each feature, male speech with white noise)

4. Conclusion

In this chapter, we have shown that it is possible to estimate the subjective speech intelligibility, as measured by the Diagnostic Rhyme Test (DRT) Chance-Adjusted percentage Correct Rate (CACR), from objective PESQ MOS-LQO scores if we have a mapping function for the noise and the phonetic feature to be tested beforehand. PESQ itself was proven to be too sensitive to noise to serve as a good scale to map to DRT CACR for wide range of signal to noise ratio. In other words, PESQ MOS-LQO saturated quickly to low scores as noise is increased, while DRT CACR stayed relatively high even with considerable noise. This suggests that PESQ may not be a good match to serve as estimation variable for DRT scores for the whole range of SNR we are interested in.

We then attempted to map the MOS-LQO of a test word to DRT CACR using polynomials trained on a training speaker, and mapped the MOS-LQO of an unknown speaker to the DRT CACR. If we use one mapping function per phonetic feature, we showed that it is possible to map the MOS-LQO to DRT CACR to some extent. However, this mapping per word generally showed a large root mean square estimation error (RMSE), mostly larger than 0.3, and the correlation between estimated and subjective DRT CACR was generally low, below 0.5 in most cases.

We then pooled the CACR of the words in each phonetic feature category to estimate the CACR for each feature, as is done in subjective testing. This was shown to dramatically decrease the error, resulting in RMSE below 0.2, and increase the correlation, to above 0.8 in most cases. This dramatic improvement was seen because the estimated CACR for the

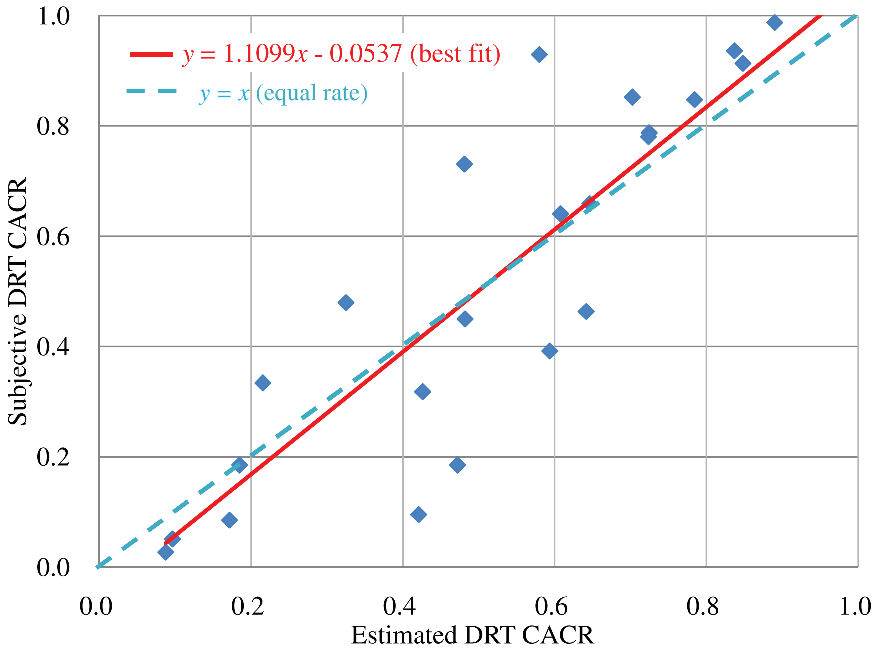


Fig. 13. Subjective CACR vs. estimated CACR (pooled for each feature, female speech with babble noise)

individual words in a phonetic feature category was evenly distributed around the subjective CACR values. By pooling all CACR, we were able to average out this deviation.

Although we have shown that it is possible to estimate the DRT CACR using PESQ-derived MOS-LQO, we still can only do so with limited accuracy. This is because PESQ itself is too sensitive to any amount of noise. Thus, we need to use the internal representation within PESQ and calculate a measure which has more linear correlation with noise levels, or we need to look at completely different objective measures. Accordingly, we have started looking at other candidate objective measures which may show higher correlation with intelligibility, *i.e.* DRT CACR. Segmental SNR and its derivatives, *e.g.* frequency-weighed segmental SNR (Hu & Loizou, 2008), seems to show much higher correlation. The composite measure proposed by the same author, which combines several objective measures, seem to be promising as well (Hu & Loizou, 2008). These measures can be mapped to DRT CACR using polynomials per phonetic feature as we have done in this paper. Preliminary results show significantly improved estimation accuracy. We plan to reveal these results in the near future in a separate paper and conference presentations.

On the other hand, we are also trying out a completely different approach to the same problem. We applied automatic speech recognizers with language models that force one of the words in the word-pair, mimicking the human recognition process of the DRT. The acoustic models were adapted to each of the speakers in the corpus, and then adapted to noise at a specified SNR. We tested with white noise, babble noise, and pseudo-speech noise. The match between subjective and estimated scores improved significantly with noise-adapted models compared to speaker-independent models and the speaker-adapted models, when

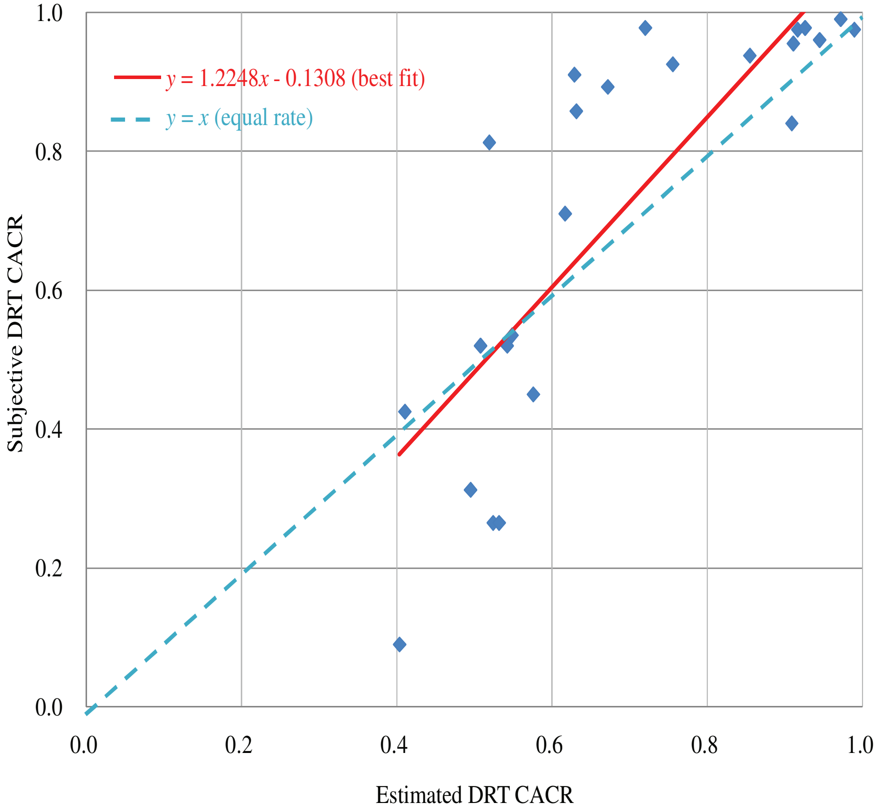


Fig. 14. Subjective CACR vs. estimated CACR (pooled for each feature, male speech with babble noise)

the adapted noise level and the tested level match. However, when SNR conditions do not match, the recognition scores degraded especially when tested SNR conditions were higher than the adapted noise level. Accordingly, we adapted the models to mixed levels of noise, *i.e.*, multi-condition training. The adapted models now showed relatively high intelligibility matching subjective intelligibility performance over all levels of noise. The correlation between subjective and estimated intelligibility scores increased to 0.94 with babble noise, 0.93 with white noise, and 0.89 with pseudo-speech noise, while the root mean square error (RMSE) reduced from more than 0.40 to 0.13, 0.13 and 0.16, respectively. Detailed results are described in a separate paper (Kondo & Takano, 2010; Takano & Kondo, 2010).

5. Acknowledgment

The work described in this chapter was supported in part by the Ministry of Education, Culture, Sports, Science and Technology Grant-in-Aid for Scientific Research (20500151), the Telecommunications Advancement Foundation, and the Research Foundation for the Electro-technology of Chubu. We also thank Professors Tetsuo Kosaka and Masaharu Kato for their assistance in the development of speaker- and noise-adapted speech models. We also thank Professor Nakagawa as well as our colleagues in the Nakagawa Laboratory for their comments and suggestions. Finally, we thank the students in the Nakagawa Laboratory and Kondo Laboratory for the daily discussions as well as their voluntary participation in the long and strenuous intelligibility tests.

6. References

Amano, S. & Kondo, K. (1999). *Lexical properties of Japanese*, Sanseido, Tokyo. in Japanese. CD publication.

ANSI (1989). Recommendation S3.2-1989: Method for measuring the intelligibility of speech over communication systems.

Beerends, J. G., Buuren, R. V., Vugt, J. V. & Verhave, J. (2009). Objective speech intelligibility measurement basis of natural speech in combination with perceptual modeling, *J. Audio Eng. Soc.* 57(5): 299–308.

Beerends, J. G., Hekstra, A. P., Rix, A. W. & Hollier, M. P. (2002). Perceptual evaluation of speech quality (PESQ) the new ITU standard for end-to-end speech quality assessment part II - psychoacoustic model, *J. Audio Eng. Soc.* 50(10): 765–778.

Fairbanks, G. (1958). Test of phonemic differentiation: The rhyme test, *J. of the Acoustical Society of America* 30: 596–600.

Fujimori, M., Kondo, K., Takano, K. & Nakagawa, K. (2006). On a revised word-pair list for the japanese intelligibility test, *Proc. International Symposium on Frontiers in Speech and Hearing Research*, Tokyo, Japan.

Hu, Y. & Loizou, P. C. (2008). Evaluation of objective quality measures for speech enhancement, *Trans. on Audio, Sp., and Lang. Process.* 16(1): 229–238.

Iida, S. (1987). On the articulation test, *J. of Acoustical Society of Japan* 43(7): 532–536. in Japanese.

ITU-T (1996). ITU-T Recommendation P.800: Method for subjective determination of transmission quality.

ITU-T (2001). ITU-T Recommendation P.862: Perceptual evaluation of quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs .

ITU-T (2003). ITU-T Recommendation P.862.1: Mapping function for transforming P.862 raw result scores to MOS-LQO.

ITU-T (2005). ITU-T Recommendation P.862.2: Wideband extension to Recommendation P.862 for the assessment of wideband telephone networks and speech codecs.

ITU-T (2007). ITU-T Recommendation G.729: Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP).

Jakobson, R., Fant, C. G. M. & Halle, M. (1952). Preliminaries to speech analysis: The distinctive features and their correlates, *Technical Report 13*, Acoustics Laboratory, MIT.

Kaga, R., Kondo, K., Nakagawa, K. & Fujimori, M. (2006). Towards estimation of Japanese intelligibility scores using objective voice quality assessment measures, *Proc. 4th Joint Meeting of the ASA and the ASJ, J. Acoust. Soc. of Am.*, Vol. 120, Honolulu, Hawaii, p. 3255.

Kitawaki, N. & Yamada, T. (2007). Subjective and objective quality assessment for noise reduced speech, *Proc. ETSI Workshop on Speech and Noise in Wideband Communication*, Vol. IV, pp. 1–4.

Kondo, K., Izumi, R., Fujimori, M., Kaga, R. & Nakagawa, K. (2007). On a two-to-one selection based Japanese intelligibility test, *J. Acoust. Soc. of Japan* 63(4): 196–205. in Japanese.

Kondo, K., Izumi, R. & Nakagawa, K. (2001). Towards a robust speech intelligibility test in Japanese, *Proc. 17th International Congress on Acoustics*, Rome, Italy, p. 7P.39.

Kondo, K. & Takano, Y. (2010). Estimation of two-to-one forced selection intelligibility scores by speech recognizers using noise-adapted models, *Proc. Interspeech*, ISCA, Makuhari, Japan, pp. 302–305.

Liu, W. M., Jellyman, K. A., Evans, N. W. D. & Mason, J. S. D. (2008). Assessment of objective quality measures for speech intelligibility, *Proc. Interspeech*, Brisbane, Australia.

Miller, G. A., Heise, G. A. & Lichten, W. (1951). The intelligibility of speech as a function of the context of the test materials, *J. of Experimental Psychology* 41: 329–335.

NHK Broadcasting Culture Research Institute (ed.) (1998). *Japanese Pronunciation Dictionary*, Japan Broadcast Publishing.

Nishimura, R., Asano, F., Suzuki, Y. & Sone, T. (1996). Speech enhancement using spectral subtraction with wavelet transform, *IEICE Trans. Fundamentals* 79-A(12): 1986–1993. in Japanese.

Rice University (1995). Signal Processing Information Base (SPIB), `http://spib.rice.edu/spib/select_noise.html`.

Rix, A. W., Hollier, M. P., Hekstra, A. P. & Beerends, J. G. (2002). Perceptual evaluation of speech quality (PESQ) the new ITU standard for end-to-end speech quality assessment part I - time-delay compensation, *J. Audio Eng. Soc.* 50(10): 755–764.

Suzuki, Y., Kondo, K., Sakamoto, S., Amano, S., Ozawa, K. & Sone, T. (1998). Perceptual tendency in word intelligibility tests by use of word-lists with controlled word familiarities, *Technical Report H-98-47*, Acoustical Society of Japan Technical Committee on Psychological and Physiological Acoustics. in Japanese.

Takano, Y. & Kondo, K. (2010). Estimation of speech intelligibility using speech recognition systems, *IEICE Trans. on Inf. and Syst.* E93-D(12): 3368–3376.

Tanaka, M. (1989). A prototype of a quality evaluation system for hearing aids, *Technical report*, Report of the Results of Research with METI Kakenhi (Grant-in-Aid). in Japanese.

Voiers, W. D. (1977). *Speech Intelligibility and Speaker Recognition*, Dowden, Hutchinson & Ross, Stroudsburg, PA, chapter Diagnostic Evaluation of Speech Intelligibility, pp. 374–387.

Voiers, W. D. (1983). Evaluating processed speech using the diagnostic rhyme test, *Speech Technology* 1: 30–39.

Spectral Properties and Prosodic Parameters of Emotional Speech in Czech and Slovak

Jiří Přibil[1] and Anna Přibilová[2]
[1]*Institute of Measurement Science, SAS,*
[2]*Faculty of Electrical Engineering & Information Technology, SUT,*
Slovakia

1. Introduction

The methods of analysis of human voice are based on the knowledge of speaker individuality. One of basic studies on speaker acoustic characteristics can be found in (Kuwabara & Sagisaka 1995). According to them the voice individuality is affected by the voice source (the average pitch frequency, the pitch contour, the pitch frequency fluctuation, the glottal wave shape) and the vocal tract (the shape of spectral envelope and spectral tilt, the absolute values of formant frequencies, the formant trajectories, the long-term average speech spectrum, the formant bandwidth). The most important factors on individuality are the pitch frequency and the resonance characteristics of the vocal tract, though the order of the two factors differs in different research studies.

According to (Scherer 2003) larynx and pharynx expansion, vocal tract walls relaxation, and mouth corners retraction upward lead to falling first formant and rising higher formants during pleasant emotions. On the other hand, larynx and pharynx constriction, vocal tract walls tension, and mouth corners retraction downward lead to rising first formant and falling higher formants for unpleasant emotions. Thus, the first formant and the higher formants of emotional speech shift in opposite directions in the frequency ranges divided by a frequency between the first and the second formant. In practice, the formant frequencies differ to some extent for different languages and their ranges are overlapped. According to (Stevens 1997) the frequency of vibration of the vocal folds during normal speech production is usually in the range 80 ÷ 160 Hz for adult males, 170 ÷ 340 Hz for adult females, and 250 ÷ 500 Hz for younger children. It means that female pitch frequencies are about twice the male pitch frequencies, pitch frequencies of younger children are about 1.5-times higher than those of females and about 3-times higher than those of males. As regards the formant frequencies, females have them on average 20 % higher than males, but the relation between male and female formant frequencies is nonuniform and deviates from a simple scale factor (Fant 2004).

Emotional state of a speaker is accompanied by physiological changes affecting respiration, phonation, and articulation. These acoustic changes are transmitted to the ears of the listener and perceived via the auditory perceptual system (Scherer 2003). From literature and our experiments follows that different types of emotions are manifested not only in prosodic patterns (F0, energy, duration) and several voice quality features (e.g. jitter, shimmer, glottal-to-noise excitation ratio, Hammarberg index) (Li et al. 2007) but also by significant changes in spectral domain (Nwe et al. 2003). Several spectral features (spectral centroid, spectral flatness measure, Renyi entropy, etc.) quantify speaker-dependent as well as emotion-dependent characteristics of a speech signal (Hosseinzadeh & Krishnan 2008). It means these features provide information which complements the vocal tract characteristics. This paper describes analysis and comparison of basic spectral properties (values and ranges of cepstral coefficients, positions of formants), complementary spectral features (spectral flatness measure), and prosodic parameters (F0 and energy, microintonation, and jitter) of male and female acted emotional speech in Czech and Slovak languages. We perform statistical analysis for four emotional states: joy, sadness, anger, and a neutral state.

2. Subject and methods

Our experiments are aimed at statistical analysis and comparison of the spectral and prosodic features in emotional and neutral speech. It comprises comparison of basic statistical parameters (minimum, maximum, mean values, and standard deviation) and calculated histograms of distribution. Extended statistical parameters (skewness, kurtosis) are subsequently calculated from these histograms and/or the histogram can be evaluated by the analysis of variances (ANOVA) approach. Hypothesis tests are used for objective classification of neutral and different emotional styles.

2.1 Evaluation of results based on statistical analysis

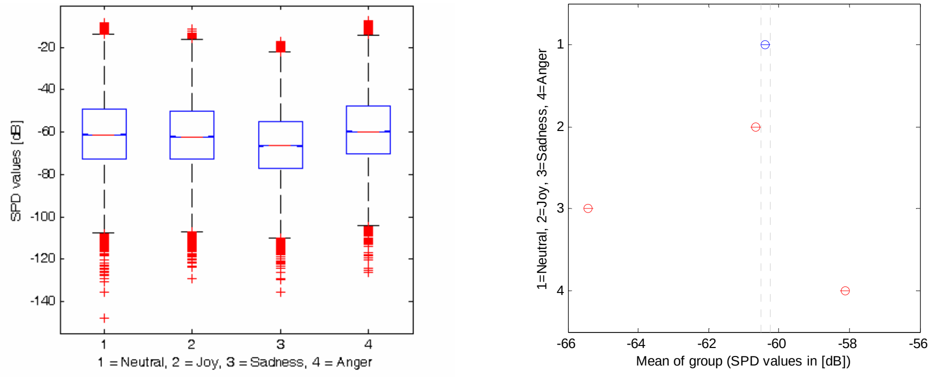
The resulting parameters obtained from our analysis experiment in the form of histograms of distribution can be applied to visual classification (determination) of speech in different emotional states (typical shapes of particular histograms), or extended statistical parameters can be subsequently calculated from these histograms for objective matching. The skewness *y* and kurtosis *k* of a distribution is defined as

$$y = \frac{E(x-\mu)^3}{\sigma^3}, \qquad k = \frac{E(x-\mu)^4}{\sigma^4}, \tag{1}$$

where μ is the mean and σ is the standard deviation of the random variable *X*, and *E*(*t*) represents the expected value of the quantity *t*. Skewness is a measure of asymmetry of the data around the sample mean. If skewness is negative, the data are spread out more to the left of the mean than to the right. If skewness is positive, the data are spread out more to the right. Kurtosis is a measure of how outlier-prone a distribution is. The kurtosis of the normal distribution is 3. Distributions that are more outlier-prone than the normal distribution have kurtosis greater than 3; distributions that are less outlier-prone have kurtosis less than 3. On the other hand, some definitions of kurtosis subtract 3 from the computed value, so that the normal distribution has kurtosis of 0 (Suhov & Kelbert 2005). We use this approach for calculation of kurtosis values in this study.

For objective comparison and matching the evaluation by ANOVA with multiple comparison of groups can be applied (Everitt 2006). This approach is more simple than recent speech recognition methods using evaluation by hidden Markov models (Srinivasan & DeLiang 2010) and is often used in other areas of biomedical research (Volaufova 2005), (Hartung et al. 2001). ANOVA gives also *F* statistics and results of the hypothesis test including probability values. Unlike the ANOVA *F* statistics, the Ansari-Bradley test (Suhov & Kelbert 2005) compares whether two independent samples come from the same distribution against the alternative that they come from distributions having the same median and shape but different variances. The result is *h* = 0, if the null hypothesis of identical distributions cannot be rejected at the 5% significance level, or *h* = 1, if the null hypothesis can be rejected at the 5% level. The hypothesis test also returns the probability of observing the given result. Small values of this probability cast doubt on the validity of the null hypothesis.

Application of described evaluation approach is demonstrated on example of the Spectral Power Density (SPD) values in [dB] of spectrograms of the sentence "*Vlak už nejede*" (Czech male speaker) uttered in neutral and three emotional styles. Fig. 1a) contains the box plot of basic statistical parameters, Fig. 1b) shows visualization multiple comparison of group means applied to the results of ANOVA statistics - each group mean is represented by a symbol and an interval around the symbol. Three means are significantly different if their intervals are disjoint, and two groups ("Neutral" and "Joy") are not significantly different if their intervals overlap. Corresponding values of Ansari-Bradley test are stored in Table 1.



a) b)

Fig. 1. Box plot of basic statistical parameters (a), visualization of multiple comparison of group means applied to the results of ANOVA statistics (b).

h/p	Neutral	Joy	Sadness	Anger
Neutral	**0/1**	0/0.309	$1/3.73\ 10^{-30}$	$1/8.66\ 10^{-55}$
Joy		**0/1**	$1/9.64\ 10^{-46}$	$1/3.14\ 10^{-25}$
Sadness			**0/1**	$1/1.05\ 10^{-169}$
Anger				**0/1**

Table 1. Results of the Ansari-Bradley hypothesis test of values corresponding to multiple comparison of group means in Fig. 1b).

2.2 Analysis and evaluation of basic spectral properties

Speech spectrum is represented very well by a pole/zero model using cepstral coefficients in comparison with linear predictive coding (LPC) corresponding only to an all-pole approximation of the vocal tract. From the input samples of the speech signal (after segmentation and weighting by a Hamming window) the complex spectrum by the Fast Fourier Transform (FFT) algorithm is calculated. In the next step the powered spectrum is computed and the natural logarithm is applied. Second application of the FFT algorithm gives the symmetric real cepstrum

$$\{c_n\} = \left\{ c_0, c_1, \ldots, c_{N_{FFT}/2} \,|\, c_{N_{FFT}/2-1}, \ldots, c_1 \right\} . \tag{2}$$

By limitation to the first N_0+1 coefficients, the Z-transform of the real cepstrum can be obtained

$$C(z) = c_0 + c_1 z^{-1} + c_2 z^{-2} + \ldots + c_{N_0} z^{-N_0}. \tag{3}$$

The truncated cepstrum represents an approximation of a log spectrum envelope

$$E(f) = c_0 + 2\sum_{n=1}^{N_0} c_n \cos(n \cdot 2\pi f). \tag{4}$$

The cepstral speech synthesis is performed by a digital filter implementing approximate inverse cepstral transformation. The system transfer function of this filter is given by an exponential relation where the exponent is the Z-transform of the truncated speech cepstrum and it represents the minimum phase approximation of the real cepstrum. Approximation of the system transfer function can be performed by a cascade connection of N_0 elementary filter structures. Using the Padé approximation of the exponential function it has been found out that the minimum number of N_0 (25/50 at 8/16 kHz sampling frequency) cepstral coefficients is necessary for sufficient approximation error (Vích 2000). As the value range of the cepstral coefficients exponentially falls, only the first eight coefficients are analyzed (the remaining coefficients practically have not influence on the filter stability, structure, and implementation).

The basic cepstral analysis scheme including the spectral features calculation is shown in the block diagram in Fig. 2. Described method of cepstral speech analysis was supplied with determination of the fundamental frequency F0 and the energy En (calculated from the first cepstral coefficient c_0). After removal of the low energy starting and ending frames by the energy threshold ($\mathrm{En}_{\mathrm{min}}$) the limited working length (number of frames) for next processing was obtained – see Fig. 3. Cepstral analysis must be preceded by classification and sorting process of the cepstral coefficients in dependence on the voice type (male / female) and the speech style (neutral / emotional). Realization of analysis of the cepstral coefficient properties was processed in following phases:

a. manual (subjective) classification of voice type and emotional speech style, further automatic processing,
b. cepstral analysis of speech signal (from the main two speech databases consisting of short utterances of male/female voice pronounced in neutral and different emotional styles).

As a graphical output, the histogram of cepstral coefficients for every emotional state was also constructed. For objective comparison, the extended statistical parameters of skewness and kurtosis were subsequently calculated. The performed statistical analysis of cepstral coefficients consists of four parts:

1. determination of basic statistical parameters of the cepstral coefficients (minimum, maximum, mean value, and standard deviation),
2. calculation and building of histograms,
3. calculation of extended statistical parameters from histograms (kurtosis and skewness),
4. comparison of the mean values and the ranges of the cepstral coefficients for emotional and neutral states.

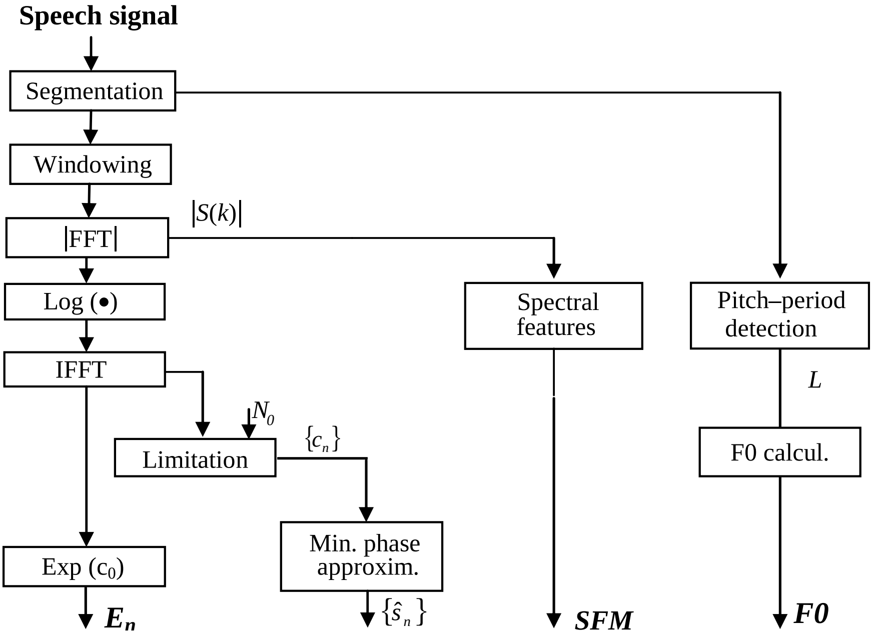


Fig. 2. Block diagram of used cepstral analysis method (N_0 = 50, and the sampling frequency f_s = 16 kHz).

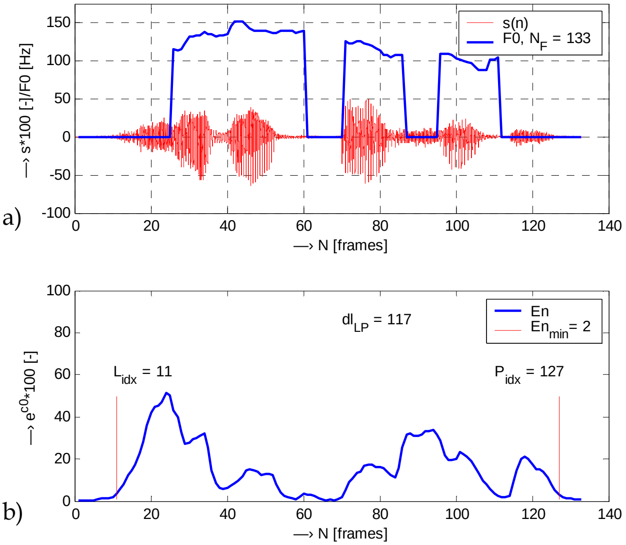


Fig. 3. The processed sentence "Život a řeč" *(Life and speech)*, Czech male speaker: speech signal with F0 contour (a), *En* contour calculated from the first cepstral coefficient c_0 (b).

In frequency domain we analyze the first three formant positions (F1, F2, and F3), and the difference between smoothed spectra for comparison of analyzed speech on segmental or phoneme level. Smoothed spectra are computed from the chosen region of interest (ROI) areas of voiced part of speech by the Welch method (Oppenheim et al. 1999). From these mean periodograms the first three formats are determined as the first three local maxima of the Welch's periodogram where its gradient changes from positive to negative.

From the summary comparison of cepstral speech analysis follows that emotional speech brings about the most significant spectral changes for voiced speech (see spectrogram in Fig. 4) therefore the extended analysis by mean periodograms of sounds was subsequently performed. For this purpose the second database consisting vowels “a”, “e”, “i”, “o”, “u” and voiced consonants “m”, “n” and “l” was used. The whole spectral analysis with the help of Welch's periodograms was practically performed in five steps:

1. calculation of smoothed spectra in the form of Welch's periodograms from the selected ROIs of voiced part of neutral and emotional speech signal,
2. determination of the first three formant positions (F1, F2, and F3) from the obtained periodograms,
3. calculation of mean emotional-to-neutral formant position ratios,
4. calculation and visual comparison of mean periodograms of sounds from the database of vowels and voiced consonants,
5. numerical matching of results from the calculated spectral distances between corresponding periodograms by the RMS method.

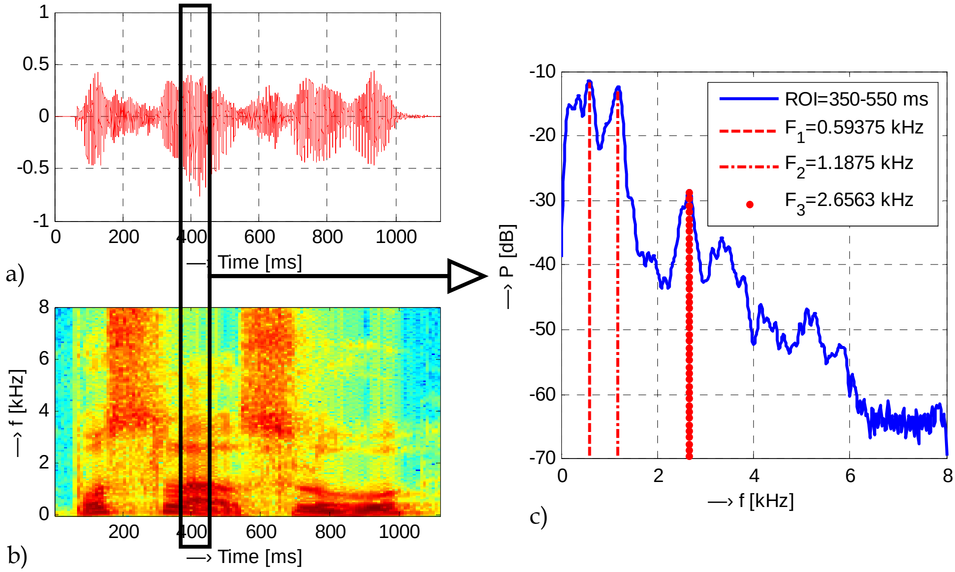


Fig. 4. Processed sentence “Poslal sluhu” (*He sent his servant*), Slovak male speaker: speech signal (a), corresponding spectrogram (b), calculated mean periodogram estimate in [dB] of selected ROI with determined formant positions F1,F2, and F3 (c).

2.3 Analysis of complementary spectral feature

As a complementary spectral feature, the spectral flatness measure (SFM) was analyzed. This spectral feature is calculated during cepstral speech analysis (see block diagram in Fig. 2) using absolute value of the fast Fourier transform denoted as $|S(k)|$

$$SFM = \frac{\left[\prod_{k=1}^{N_{FFT}/2} \left|S(k)\right|^2\right]^{\frac{2}{N_{FFT}}}}{\frac{2}{N_{FFT}} \sum_{k=1}^{N_{FFT}/2} \left|S(k)\right|^2}. \tag{5}$$

According to psychological research of emotional speech different emotions are accompanied by different spectral noise (Scherer et al. 2003). In cepstral speech synthesis the spectral flatness measure SFM was used to determine voiced/unvoiced energy ratio in voiced speech analysis (Vích 2000). The SFM values lie generally in the range of (0 ÷ 1) – the zero value represents totally voiced signal (for example pure sinusoidal signal); in the case of SFM = 1, the totally unvoiced signal is classified (for example white noise signal). According to the statistical analysis of the Czech and Slovak words the ranges of SFM = (0 ÷ 0.25) for voiced speech frames and SFM = (0 ÷ 0.75) for unvoiced frames were evaluated (Madlová & Přibil 2000). The demonstration example in Fig. 5 shows the input speech signal with detected pitch frequency F0 and calculated SFM values with voiceness classification.

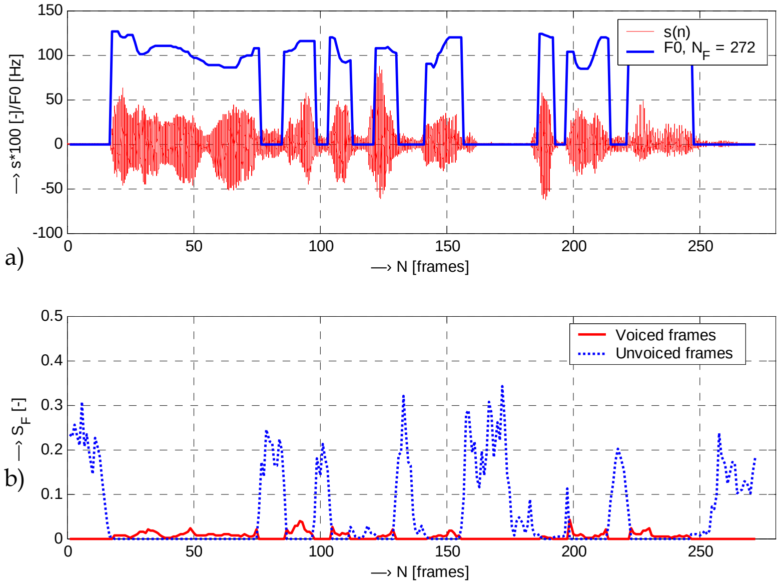


Fig. 5. Demonstration of SFM calculation: input speech signal – sentence "Lenivý si a zle gazduješ" (*You are lazy and you keep your house ill*) pronounced in angry emotional style, male Slovak speaker with F0 contour (a), SFM values for voiced and unvoiced frames (b).

In our algorithm, the values of SFM are obtained from the voiced speech frames and are separately processed in dependence on voice type (male / female). For every voice type the SFM values were subsequently sorted by emotional styles and stored in separate stacks. These classification operations were performed manually, by subjective listening method. Next operations with the stacks were performed automatically – calculation of statistical parameters: minimum, maximum, mean values, and standard deviation. From the mean spectral feature values the ratio between emotional and neutral states is subsequently calculated. As a graphical output used for visual comparison (subjective method) the histogram of sorted spectral features values for each of the stacks is also calculated. Consequently the extended statistical parameters of histograms (skewness and kurtosis) were subsequently calculated. The second approach based on ANOVA was applied to SFM values together with multiple comparison of groups test as an objective evaluation method– see block diagram in Fig. 6.

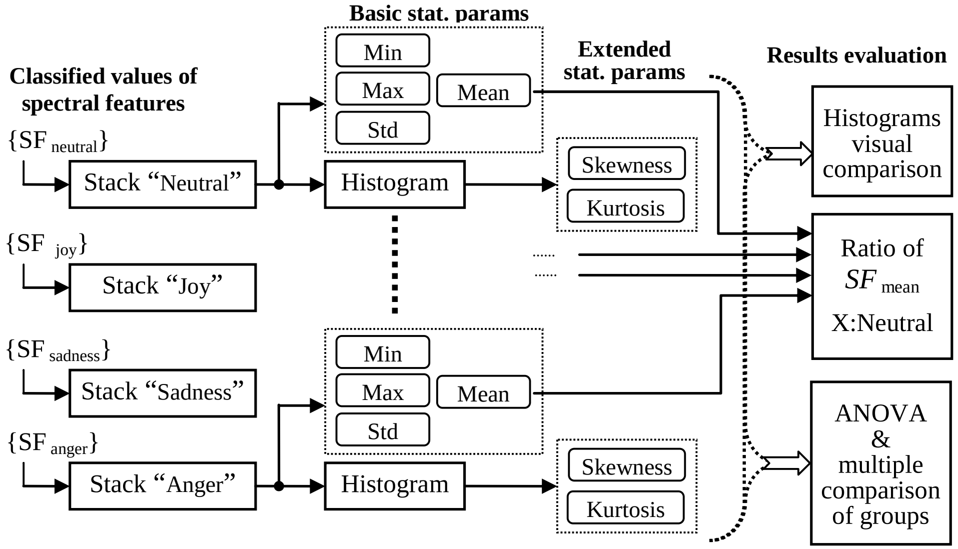


Fig. 6. Block diagram of processed operations with the stacks filled with classified spectral feature values.

2.4 Analysis of prosodic parameters and their comparison

Melody of speech utterances is given by the fundamental frequency (F0) contour. Microintonations as well as jitter together with the sentence melody and the word melody also represent the speech melody. Microintonation can be supposed to be a random, band-pass signal described by statistical parameters.

The whole prosodic parameter analysis procedure is divided into four phases:

1. analysis of the speech signal: determination of F0 and energy contour,
2. analysis of F0 contour, microintonation extraction, determination of pitch periods in the voiced parts of the speech signal – see example in Fig. 7,
3. statistical evaluation of F0, energy, microintonation, zero crossings, and calculation of ratios for emotional / neutral states (see block diagram in Fig. 8),

4. microintonation signal spectral analysis and 3-dB bandwidth (B_3) determination from the concatenated signal.

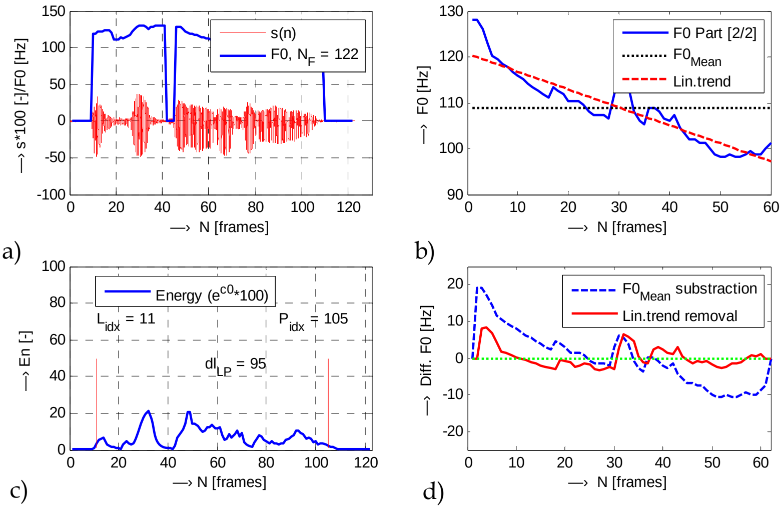


Fig. 7. Demonstration of microintonation analysis: speech signal with F0 contour (a), the second voiced part: original F0, mean F0, and LT (b), energy contour (c), differential signal after $F0_{mean}$ and LT subtraction (d) – the sentence “Řekl Radomil” *(Radomil said)* uttered by a male Czech speaker.

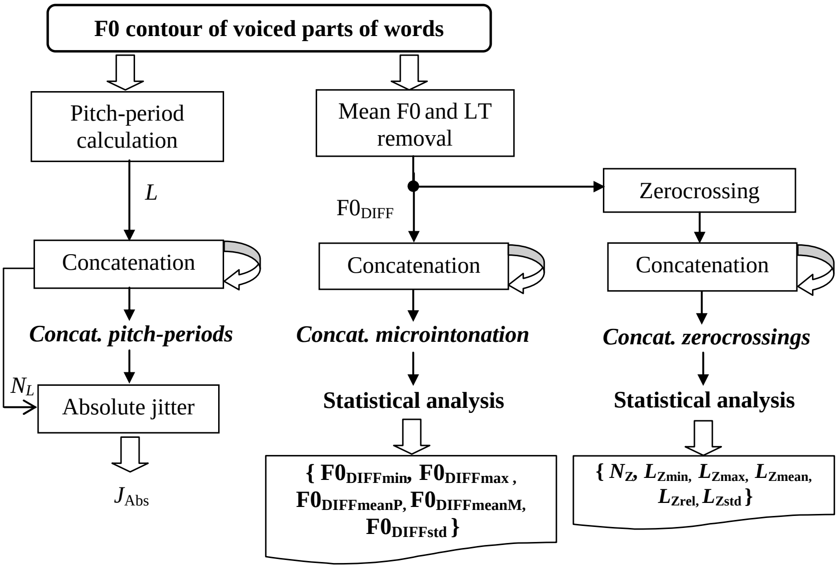


Fig. 8. Block diagram of microintonation signal analysis including basic and zero crossing statistical analysis.

The introductory microintonation processing phase consists of the following steps:

1. determination of the melody contours from the voiced parts of speech smoothed by a median filter,
2. determination of F0 mean values, $F0_{range}$ (as difference of minimum and maximum) and calculation of the linear trend (LT) by the mean square method,

$$LT\ (a,b) = a + b\ n, \tag{6}$$

where $n = 1,2,...N_F$ and N_F is number of frames of the F0 contour. The best linear fit to a given set of F0 values is solved by least squares fitting technique of linear regression yielding

$$a = \frac{\sum_{n=1}^{N_F} F0(n) \sum_{n=1}^{N_F} n^2 - \sum_{n=1}^{N_F} n \sum_{n=1}^{N_F} n\ F0(n)}{N_F \sum_{n=1}^{N_F} n^2 - \left(\sum_{n=1}^{N_F} n\right)^2}, \quad b = \frac{N_F \sum_{n=1}^{N_F} n\ F0(n) - \sum_{n=1}^{N_F} n \sum_{n=1}^{N_F} F0(n)}{N_F \sum_{n=1}^{N_F} n^2 - \left(\sum_{n=1}^{N_F} n\right)^2}, \tag{7}$$

3. calculation of differential microintonation signal $F0_{DIFF}$ by subtraction of these values from the corresponding F0 contours ($F0_{mean}$ and LT removal) – see Fig. 7b)

$$F0_{DIFF}(n) = \left(F0(n) - F0_{mean}\right) - LT(n), \tag{8}$$

4. calculation of the absolute jitter values J_{Abs}, as the average absolute difference between consecutive pitch periods L measured in samples (Farrús et al. 2007)

$$J_{Abs} = \frac{1}{f_s(N_L - 1)} \sum_{n=1}^{N_L - 1} \left| L_n - L_{n+1} \right|, \tag{9}$$

where f_s is the sampling frequency and N_L is the number of extracted pitch periods,

5. detection of zero crossings, calculation of zero crossing periods L_Z.

Spectral analysis of concatenated differential microintonation signal is also carried out for all emotions. This analysis phase is divided into three steps:

1. Calculation of the frequency parameters from the zero crossing periods $L_{Zx} = \{L_{Zmin}, L_{Zmax}, L_{Zmean}, L_{Zrel}, L_{Zstd}\}$ as $F_{Zxl} = f_F / (2\ L_{Zx})$, where f_F is the frame frequency.
2. Microintonation signal spectral analysis by periodogram averaging using the Welch method.
3. Determination of B_3 values from these spectra for each of the emotion types.

To obtain spectrum of smoothed microintonation signal (see Fig. 9b), the concatenated differential F0 signal is filtered by a moving average (MA) filter of the length M_F (voiced parts shorter than M_F+2 frames are not processed in further analysis) – see Fig. 9a).

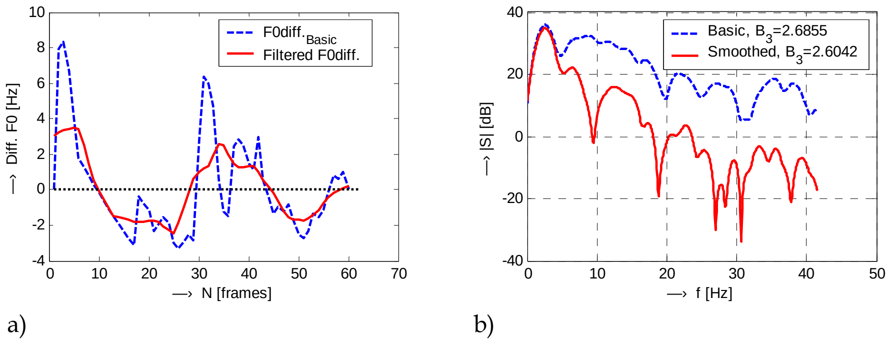


Fig. 9. Demonstration of microintonation smoothing and spectrum determination (obtained from the same sentence as in Fig. 7): basic differential F0 signal and the one filtered by moving average (a), corresponding spectra and their 3-dB bandwidths B_3 (b).

3. Material, experiments and results

As follows from our previous experiments, the basic spectral properties (cepstral coefficients and formant positions) as well as the complementary spectral features depend on a speaker but they do not depend on nationality (it was confirmed that it holds for the Czech and Slovak languages). Therefore, the created speech database consists of neutral and emotional sentences uttered by several speakers (extracted from the Czech and Slovak stories performed by professional actors). The speech material was collected in two databases (separately from male – 134 sentences, and female voice – 132 sentences, 8+8 speakers altogether) consisting of sentences with duration from 0.5 to 5.5 seconds, resampled at 16 kHz representing four emotional states (sad, joyful, angry, and neutral for comparison). Classification of emotional states was carried out manually by subjective listening method.

The F0 vales (pitch contours) were given by autocorrelation analysis method (Oppenheim 1999) with experimentally chosen pitch-ranges by visual comparison of testing sentences (one typical sentence from each of emotions and voice classes) as follows: 35÷250 Hz for male, and 105÷350 Hz for female voices. The F0 values were next compared and corrected by results obtained with the help of the PRAAT program (Boersma & Weenink 2008) with similar internal settings of F0 values.

Speech signal analysis was performed for total number of 25988 frames (8 male speakers) and 24017 frames (8 female speakers). The formant positions and the spectral flatness values were determined only for the voiced frames (totally 11639 of male and 13464 of female voice). In the case of prosodic parameters analysis, the minimum length of the processed voiced parts was set to 10 frames and the corresponding length of $M_F = 8$ for moving average filter was chosen. Number of analyzed voiced parts / voiced frames) was in total:

a. Male: neutral - 112/2698, joy - 79/1927, sadness - 128/3642, anger - 104/ 2391.
b. Female: neutral - 86/2333, joy - 87/2541, sadness - 92/2203, anger - 91/2349.

3.1 Results of analysis of basic spectral properties

Results of determined basic statistical parameters of the first 8 cepstral coefficients for different speech styles are shown in the form of box plot graph in Fig. 10 (male voice).

Summary histograms of cepstral coefficients (c_1-c_8) are shown in Fig. 11 and comparison of histogram contours for different emotions of cepstral coefficients (c_1-c_4) is shown in Fig. 12 (both male voice). Table 2 contains values of kurtosis parameters and Table 3 contains values of skewness obtained from the compared histograms of c_1-c_4 (for male and female voices).

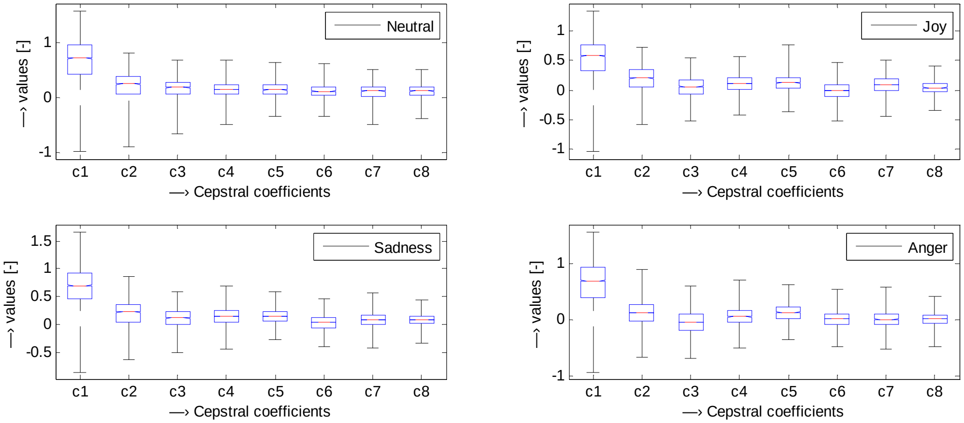


Fig. 10. Box plot of basic statistical parameters of the first 8 cepstral coefficients for different speech styles - male voice.

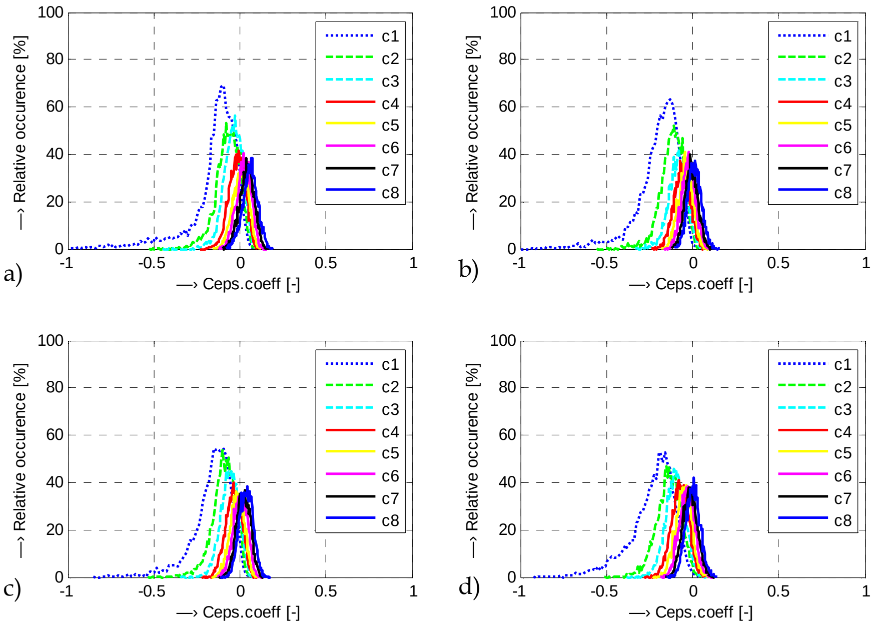


Fig. 11. Histograms of the first 8 cepstral coefficients for different speech styles (male voice): "neutral" speech (a), "joy" (b), "sadness" (c), and "anger" (d).

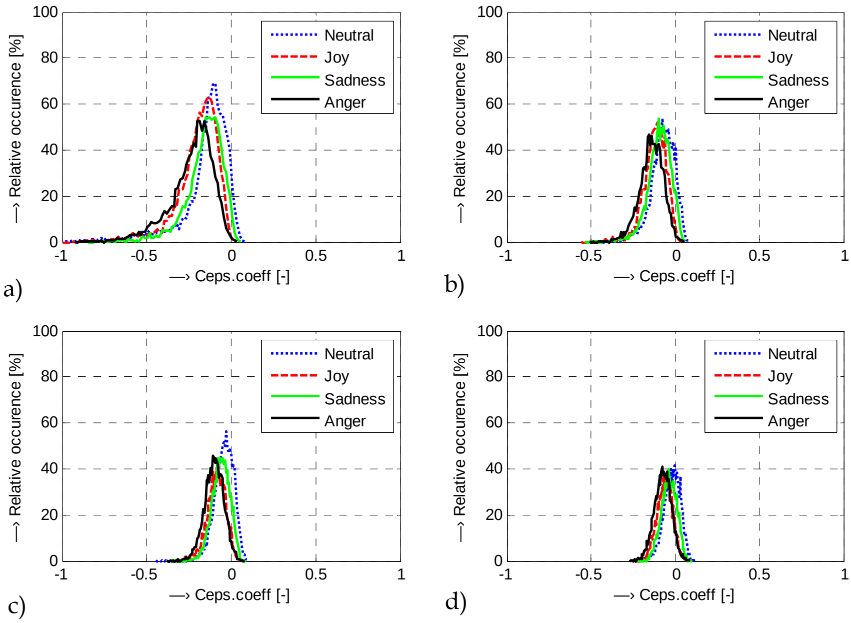


Fig. 12. Histogram comparison for different speech styles (male voice): for cepstral coefficients c_1 (a), coefficients c_2 (b), coefficients c_3 (c), and coefficients c_4 (d).

Emotion	**Male voice**				**Female voice**			
	c_1	c_2	c_3	c_4	c_1	c_2	c_3	c_4
Neutral	3.93	1.36	1.17	0.65	7.82	5.86	1.42	-0.26
Joy	2.53	0.91	0.31	0.01	6.16	3.86	0.11	-0.09
Sadness	1.72	0.82	0.29	-0.06	4.03	2.81	-0.09	-0.24
Anger	1.12	0.01	0.11	0.04	2.78	1.45	0.11	-0.04

Table 2. Kurtosis parameters determined from histograms of c_1-c_4 cepstral coefficients for male and female voices.

Emotion	**Male voice**				**Female voice**			
	c_1	c_2	c_3	c_4	c_1	c_2	c_3	c_4
Neutral	-1.79	-0.99	-0.93	-0.73	-2.47	-1.54	-0.63	-0.14
Joy	-1.20	-0.64	-0.42	-0.22	-2.00	-1.65	-0.33	-0.21
Sadness	-1.03	-0.75	-0.46	-0.13	-1.64	-1.45	-0.26	-0.13
Anger	-0.84	-0.36	-0.12	0.09	-1.42	-1.09	-0.29	-0.16

Table 3. Skewness parameters determined from histograms of c_1-c_4 cepstral coefficients for male and female voices.

Results of basic statistical parameters for the first three formant positions F1, F2, and F3 of male and female voice in neutral speech are shown are shown in Fig. 13, detailed histograms of distribution are shown in Fig. 14. Comparison of histograms of F1, F2, and F3 for different speech styles are introduced in Fig. 15 (male voice) and Fig. 16 (female voice). Table 4 contains values of kurtosis parameters and Table 5 contains values of skewness obtained from the compared histograms of F1, F2, and F3 (for male and female voices). Summary results of mean neutral-to-emotional formant position ratios for both voices can be seen in Table 6.

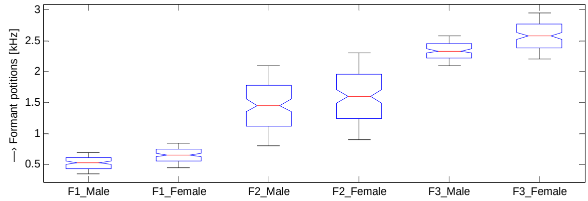


Fig. 13. Box plot of basic statistical parameters of analysis of the first three format positions (male and female voice, neutral speech style).

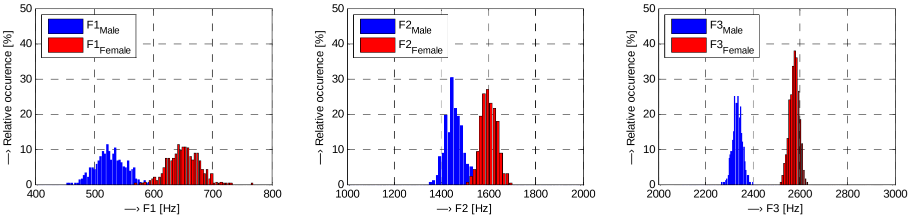


Fig. 14. Detailed histograms of the first three format positions – male and female voice, neutral speech style.

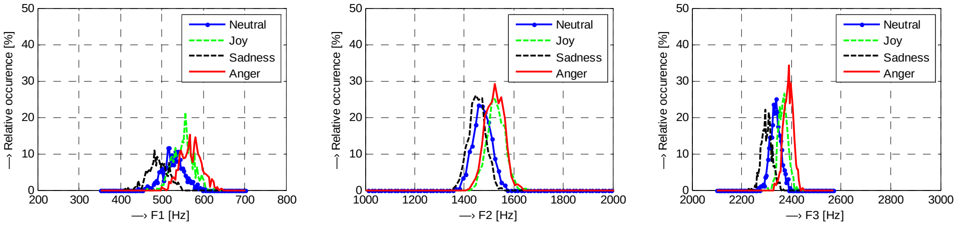


Fig. 15. Comparison of histograms of the first three formant positions for different speech styles – male voice.

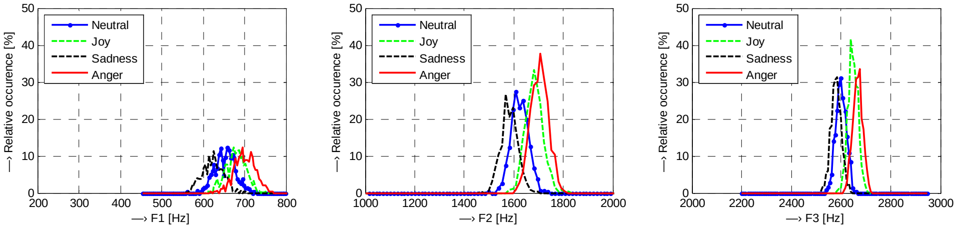


Fig. 16. Comparison of histograms of the first three formant positions for different speech styles – female voice.

Emotion	F_1 male	F_2 male	F_3 male	F_1 female	F_2 female	F_3 female
Neutral	2.702	9.257	4.884	2.771	11.440	9.823
Joy	3.298	10.457	5.253	2.613	12.155	10.376
Sadness	2.095	10.655	5.406	2.594	13.919	10.257
Anger	1.694	8.263	5.389	2.054	11.121	10.476

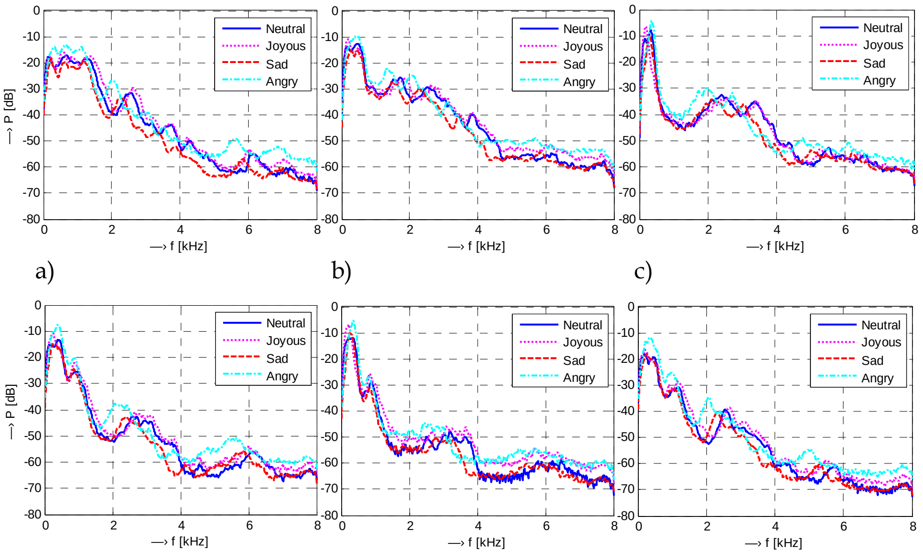
Table 4. Kurtosis parameters determined from histograms of the first three formant positions for male and female voices.

Emotion	F_1 male	F_2 male	F_3 male	F_1 female	F_2 female	F_3 female
Neutral	-1.081	0.207	-0.554	-0.999	0.532	0.266
Joy	-1.044	0.372	-0.439	-1.029	0.596	0.369
Sadness	-1.195	0.404	-0.458	-1.101	0.777	0.352
Anger	-1.297	0.100	-0.435	-1.141	0.471	0.349

Table 5. Skewness parameters determined from histograms of the first three formant positions for male and female voices.

Formant ratio	$F_{1\text{male}}$	$F_{2\text{male}}$	$F_{3\text{male}}$	$F_{1\text{female}}$	$F_{2\text{female}}$	$F_{3\text{female}}$
Joyous: neutral	0.712	1.025	1.038	0.898	1.082	1.049
Sad: neutral	1.043	0.813	0.899	1.353	0.948	0.938
Angry: neutral	1.123	0.795	0.762	1.282	0.885	0.887

Table 6. Summary results of mean emotional-to- neutral formant position ratios.



a) b) c)

d) e) f)

Fig. 17. Mean periodograms of analyzed voiced speech parts corresponding to sounds: "a" (a), "e" (b), "i" (c), "o" (d), "u" (e), and "l" (f) – male voice.

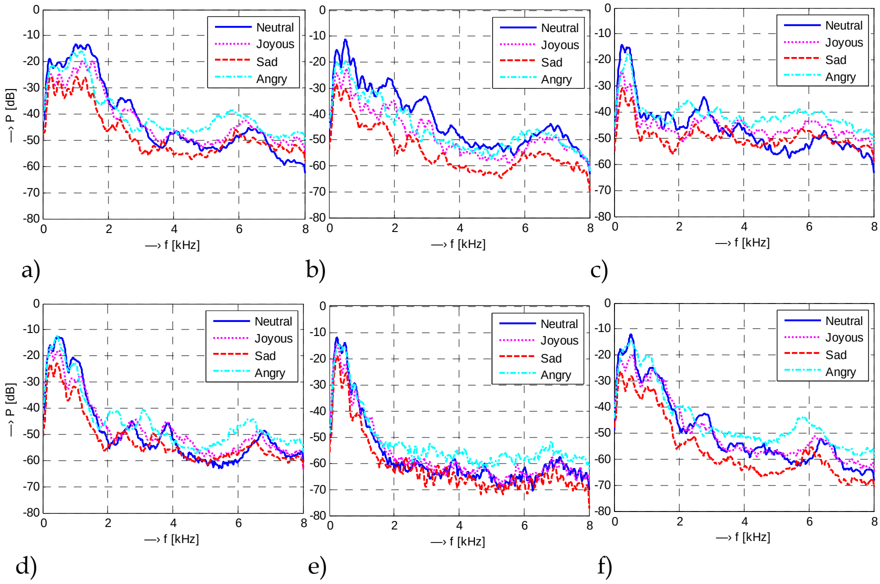


Fig. 18. Mean periodograms of analyzed voiced speech parts corresponding to sounds: "a" (a), "e" (b), "i" (c), "o" (d), "u" (e), and "l" (f) – female voice.

Results of extended analysis of formant positions by Welch's periodograms of sounds from the database of vowels and voiced consonants (detailed periodograms corresponding to sounds "a", "e", "i", "o", "u" and "l") are shown in Fig. 17 (male voice) and Fig. 18 (female voice). The spectral distances calculated between mean periodograms in "neutral" and emotional styles are summarized in Table 7.

Neutral – to:	Male voice			Female voice		
	joyous	sad	angry	joyous	sad	angry
D_{RMS} of "a" [dB]	2.517	4.516	5.845	4.598	7.223	8.382
D_{RMS} of "e" [dB]	2.608	3.551	4.862	5.708	6.599	13.012
D_{RMS} of "i" [dB]	2.427	3.769	5.279	5.794	7.236	8.060
D_{RMS} of "o" [dB]	2.710	3.639	6.110	3.841	5.866	6.370
D_{RMS} of "u" [dB]	3.509	4.596	5.846	2.692	4.973	6.298
D_{RMS} of "m"[dB]	2.205	4.809	6.595	4.186	4.724	4.774
D_{RMS} of "n"[dB]	2.160	3.852	4.615	3.985	6.597	8.909
D_{RMS} of "l" [dB]	2.839	3.408	6.063	3.207	6.929	8.064

Table 7. Summary results of spectral distances of analyzed sounds (D_{RMS} are calculated between periodograms of "neutral" and emotional styles) for male and female voices.

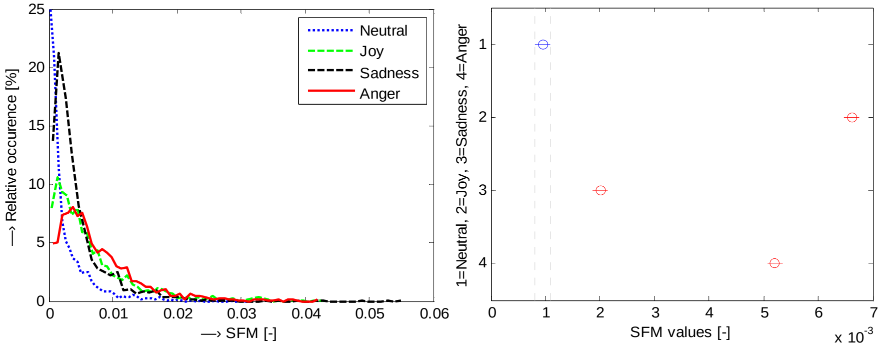
3.2 Results of analysis of a complementary spectral feature

The results of basic statistical parameters of the spectral flatness values for male and female voice analysis determined only from the voiced frames are summarized in Table 8. Histograms of *SFM* values for different emotions together with visualization of the difference between group means calculated using ANOVA statistics are shown in Fig. 19 (male voice) and Fig. 20

(female voice). Corresponding values of Ansari-Bradley test are stored in Table 9 (male voice) and Table 10 (female voice). The main result – mean spectral flatness value ratios between different emotional states and a neutral state – is given in Table 11.

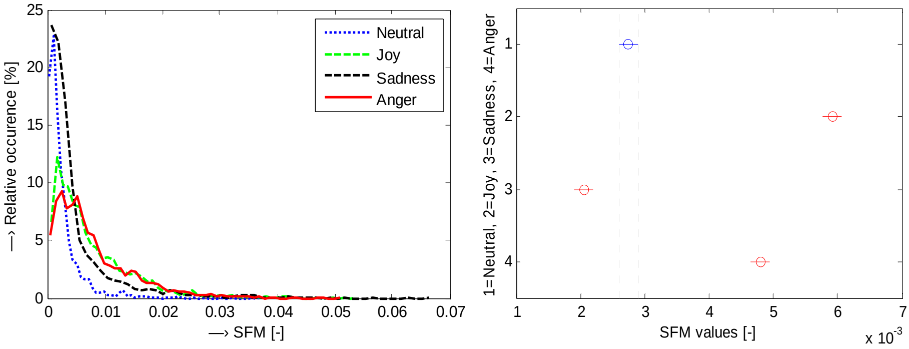
Emotion	**Male voice**				**Female voice**			
	mean	**min**	**max**	**std**	**mean**	**min**	**max**	**std**
Neutral	0.00286	$3.78\cdot10^{-5}$	0.03215	0.00364	0.00274	$3.15\cdot10^{-5}$	0.03731	0.00346
Joy	0.00662	$1.36\cdot10^{-4}$	0.04327	0.00650	0.00784	$2.07\cdot10^{-4}$	0.05414	0.00726
Sadness	0.00444	$1.12\cdot10^{-4}$	0.05540	0.00462	0.00506	$9.48\cdot10^{-5}$	0.06694	0.00674
Anger	0.00758	$2.28\cdot10^{-4}$	0.04228	0.00614	0.00807	$1.41\cdot10^{-4}$	0.05129	0.00692

Table 8. Summary results of basic statistical analysis of the spectral flatness values for male and female voice, voiced frames.



a) b)

Fig. 19. Histograms of SFM values for different speech styles (a), the difference between group means with the help of ANOVA statistics (b) - male voice, voiced frames.



a) b)

Fig. 20. Histograms of SFM values for different speech styles (a), the difference between group means with the help of ANOVA statistics (b) - female voice, voiced frames.

h/p	Neutral	Joy	Sadness	Anger
Neutral	**0/1**	$1/3.60\ 10^{-20}$	1/0.002	$1/2.10\ 10^{-11}$
Joy		**0/1**	$1/3.95\ 10^{-46}$	$1/1.21\ 10^{-35}$
Sadness			**0/1**	$1/1.52\ 10^{-39}$
Anger				**0/1**

Table 9. Results of the Ansari-Bradley hypothesis test of SFM values corresponding to Fig. 19b).

h/p	Neutral	Joy	Sadness	Anger
Neutral	**0/1**	$1/2.78\ 10^{-12}$	1/0.0015	$1/8.34\ 10^{-9}$
Joy		**0/1**	$1/5.57\ 10^{-26}$	$1/2.08\ 10^{-15}$
Sadness			**0/1**	$1/4.27\ 10^{-11}$
Anger				**0/1**

Table 10. Results of the Ansari-Bradley hypothesis test of SFM values corresponding to Fig. 20b).

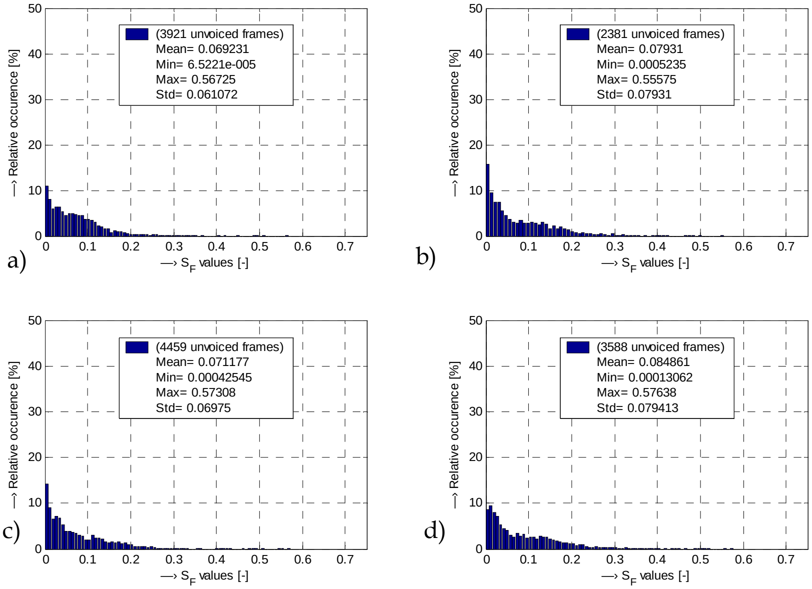


Fig. 21. Histograms of spectral flatness values calculated from the unvoiced frames (male voice): "neutral" style (a), and emotions - "joy" (b), "sadness" (c), and "anger" (d).

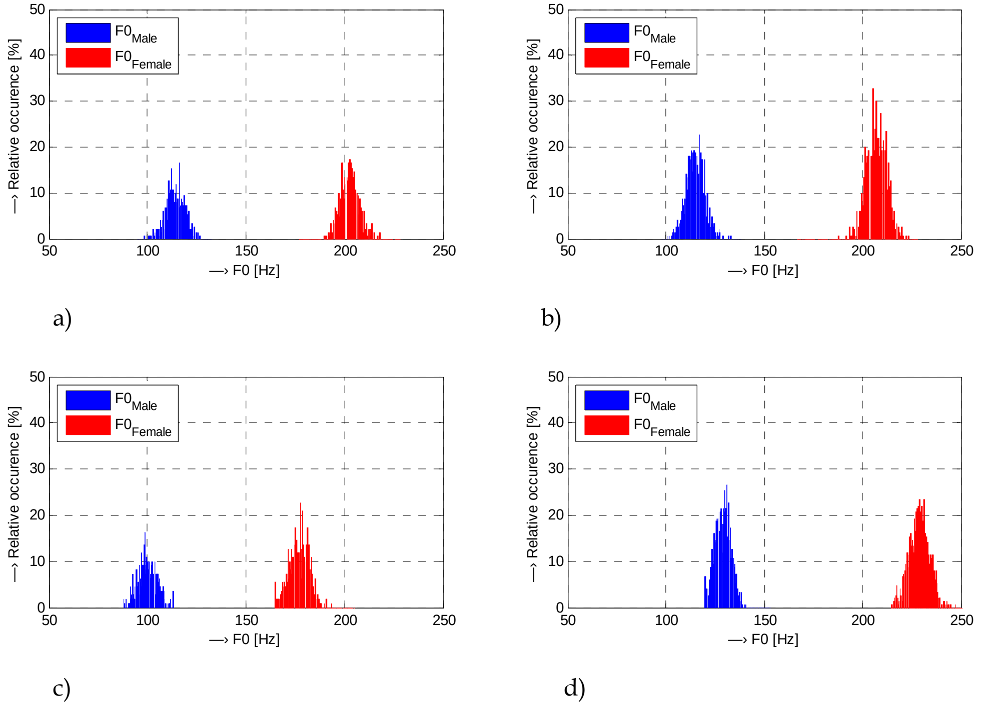
mean *SFM* **ratio**	**joy**:neutral	**sadness**:neutral	**anger**:neutral
male voice	1.349	1.725	1.321
female voice	1.455	1.795	1.377

Table 11. Mean spectral flatness values ratios between different emotional states and a neutral state (for voiced frames only).

3.3 Results of analysis of prosodic parameters

F0 contour was created from the frames with energy exceeding a chosen threshold. Histograms of F0 values distribution are shown in Fig. 22, the basic statistical parameters of $F0_{DIFF}$ and L_Z values for male and female voices of neutral and emotional speech as the box plots are presented in Fig. 23. Results of statistical analysis of energy contours (calculated from the first cepstral coefficient c_0) are shown in Fig. 24 and Table 12 consists of energy and absolute jitter ratios for emotional speech styles and neutral style for both voices. In Table 13, there are stored together mean $F0_{DIFF}$ values and absolute jitter values. Resulting summary emotional-to-neutral ratios of mean $F0_{DIFF}$ and $F0_{RANGE}$ for male and female voice are in Table 14.

Results of basic statistical analysis of zero crossing periods L_Z are shown in Table 15. For objective matching of L_Z the ANOVA and multiple comparison of group means together with the Ansari-Bradley test were performed – see Fig. 25 and results in Tables 16 to 18. Zero crossing periods were next used to calculate microintonation signal spectral analysis. Summary results including the 3-dB bandwidth values for male and female voices are shown in Table 19. The average microintonation spectra can be seen in Fig. 26 (male voice) and Fig. 27 (female voice).



a)

b)

c)

d)

Fig. 22. Histograms of F0 values for male and female voices in "neutral" style (a), and emotions "joy" (b), "sadness" (c), and "anger" (d).

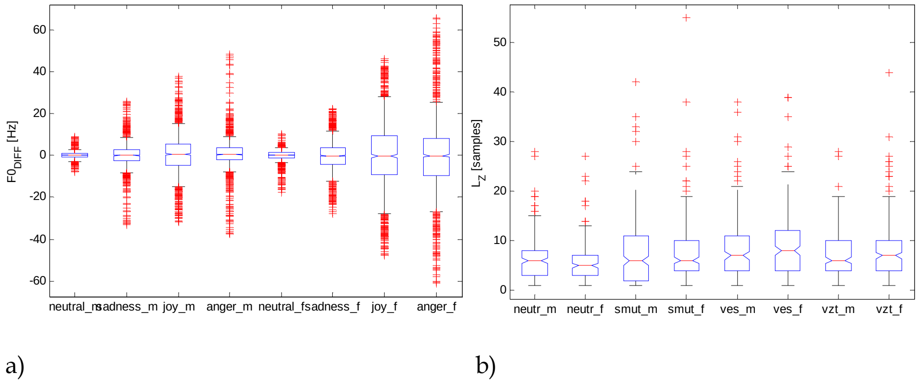


Fig. 23. Box plot of basic statistical parameters of F0$_{DIFF}$ (a) and L_Z values (b) for male and female voices.

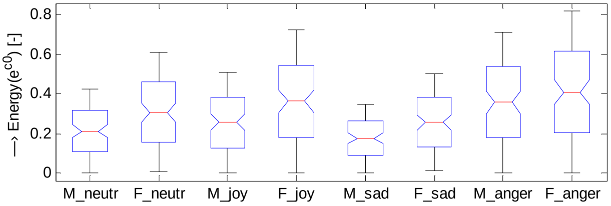


Fig. 24. Results of statistical analysis of energy contours (calculated from the first cepstral coefficient c_0): male / female voice, neutral / emotional states.

Voice	**energy ratio X:neutral**			**J_{Abs} ratio X:neutral**		
	joy	**sadness**	**anger**	**joy**	**sadness**	**anger**
Male	1.32	0.95	1.70	2.45	1.55	2.77
Female	1.50	0.73	1.84	1.94	1.41	2.06

Table 12. Summary male and female energy and absolute jitter ratios between different emotional states and a neutral state.

Emotion	**F0$_{DIFFmean}$ male**	**F0$_{DIFFmean}$ female**	**J_{Abs} male**	**J_{Abs} female**
Neutral	2.66	3.67	0.29	0.17
Joy	7.27	8.49	0.71	0.33
Sadness	4.02	6.29	0.45	0.24
Anger	8.62	10.16	0.60	0.35

Table 13. Mean values of differential F0 in [Hz] (calculated from positive microintonation values) together with absolute jitter values (in [ms]).

F0 ratio	**F0$_{\mathrm{mean}}$ joy**	**F0$_{\mathrm{mean}}$ sadness**	**F0$_{\mathrm{mean}}$ anger**	**F0$_{\mathrm{range}}$ joy**	**F0$_{\mathrm{range}}$ sadness**	**F0$_{\mathrm{range}}$ anger**
male voice	1.18	0.81	1.16	1.25	0.62	1.30
female voice	1.32	0.79	1.27	1.52	0.65	1.68

Table 14. Summary male and female F0 parameters modification ratio values between emotional and neutral speech.

Emotion	**Male voice**			**Female voice**		
	L_{Zmax}	L_{Zmean}	L_{Zstd}	L_{Zmax}	L_{Zmean}	L_{Zstd}
Neutral	57	6.82	5.69	40	6.64	5.23
Joy	23	6.74	4.57	28	5.26	3.78
Sadness	59	8.26	6.52	40	6.69	5.43
Anger	26	6.04	4.19	30	6.32	4.43

Table 15. Summary results of zero crossing basic statistical analysis (zero crossing period L_Z parameters in [frames]) – male and female voice, L_{Zmin} = 1.

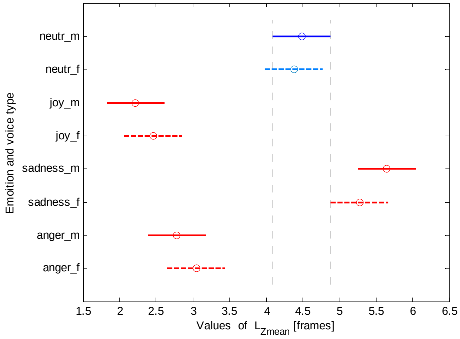


Fig. 25. Graphical results of zero crossing periods L_Z multiple comparison of ANOVA (male and female voice groups) for corresponding emotions.

h / p	**Neutral**	**Joy**	**Sadness**	**Anger**
Neutral	**0/1**	1 / 3.37 10^{-7}	1 / 0.035	1 / 0.066
Joy		**0/1**	1 / 8.99 10^{-7}	1 / 0.006
Sadness			**0/1**	1 / 0.021
Anger				**0/1**

Table 16. Partial results of zero crossing periods L_Z Ansari-Bradley hypothesis test based on comparison of distributions – male voice group.

h / p	**Neutral**	**Joy**	**Sadness**	**Anger**
Neutral	0 / 1	1 / 4.8810^{-15}	1 / 0.006	1 / 0.017
Joy		**0/1**	1 / 0.002	1 / 4.01.10^{-8}
Sadness			**0/1**	1 / 0.002
Anger				**0/1**

Table 17. Partial results of zero crossing periods L_Z hypothesis test – female voice group.

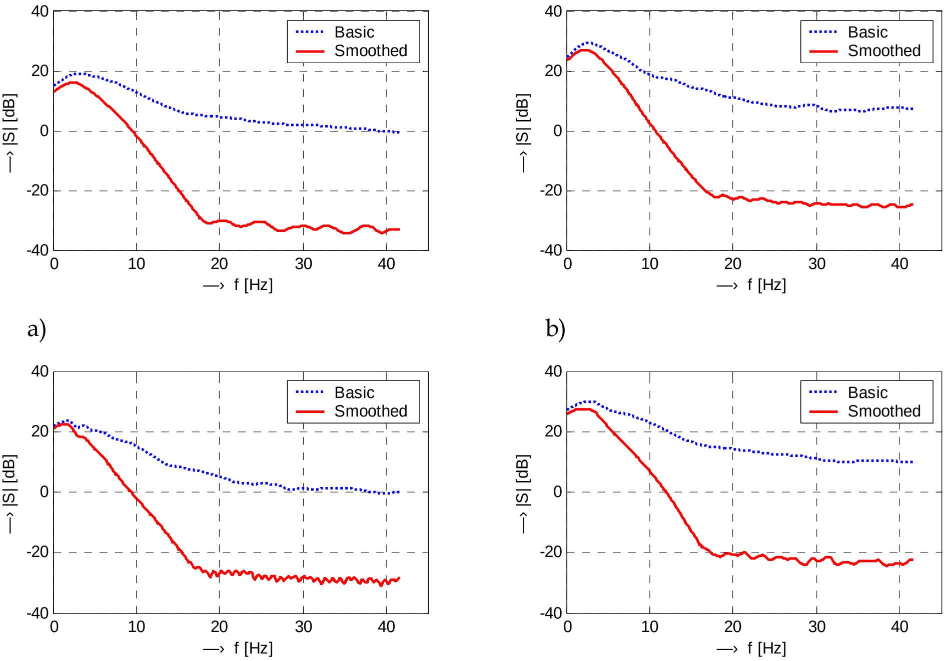
	Neutral	Joy	Sadness	Anger
h / p	0 / 0.4397	0 / 0.8926	0 / 0.6953	0 / 0. 5773

Table 18. Summary results of zero crossing periods L_Z hypothesis test (comparison male vs. female voice group) between particular emotions.

Emotion	**Male voice**					**Female voice**				
	F_{Zmin}1)	F_{Zmean}	F_{Zrel}	B_3	B_{3F} 2)	F_{Zmin}1)	F_{Zmean}	F_{Zrel}	B_3	B_{3F} 2)
Neutral	1.60	6.89	8.83	6.75	4.56	2.23	11.88	14.60	11.59	6.71
Joy	0.71	5.04	6.45	4.56	3.82	1.56	9.41	11.94	9.03	5.61
Sadness	0.73	6.11	7.78	4.39	2.69	1.56	9.33	11.66	7.20	3.17
Anger	1.81	6.18	8.00	5.37	4.07	2.08	9.88	12.59	10.74	5.86

1) L_{Zmin} = 1 ⇨ F_{Zmax}= f_F / 2
2) 3-dB bandwidth for signal smoothed by MA filter with M_F = 8

Table 19. Summary results of spectral analysis (frequency parameters in [Hz] derived from concatenated differential F0 signal) – male and female voice.



a)

b)

c)

d)

Fig. 26. Spectra of microintonation used for 3-dB bandwidth determination for emotions (with and without smoothing by moving average): “neutral” (a), “sadness” (b), “joy” (c), and “anger” (d) - male voice.

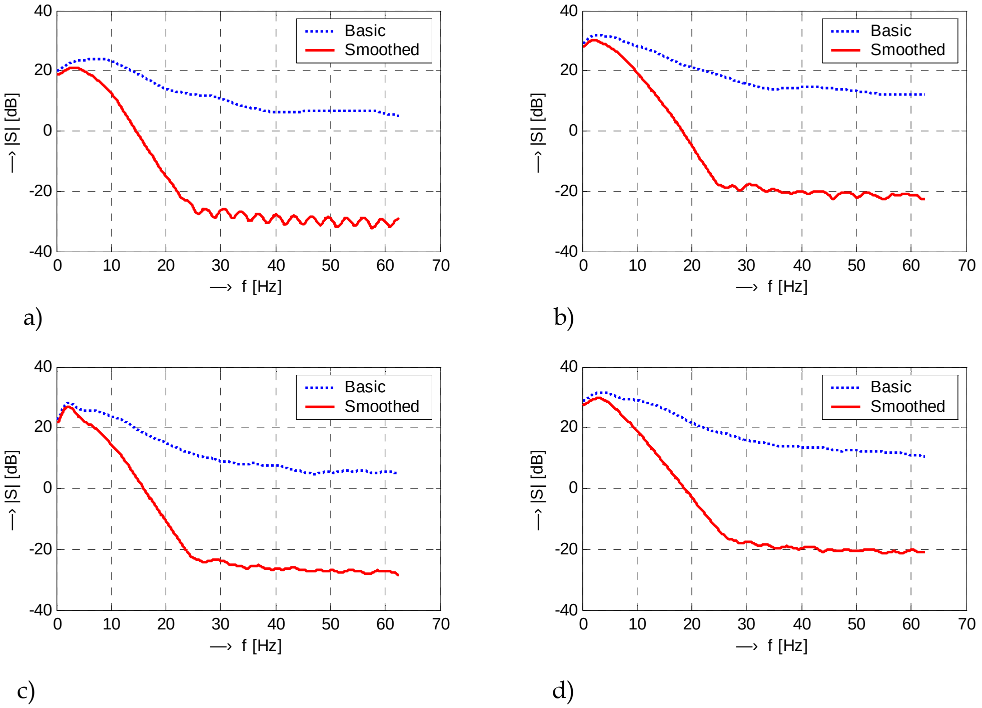


Fig. 27. Spectra of microintonation used for 3-dB bandwidth determination for emotions (with and without smoothing by moving average): “neutral” (a), “sadness” (b), “joy” (c), and “anger” (d) - female voice.

4. Discussion and conclusion

Results of performed analysis and comparison (consequently computed parameter ratios between emotional and neutral states) will be applied for extension of the text-to-speech (TTS) system enabling expressive speech production of voices (male / female) or it be also used in emotional speech transformation (conversion) method based on cepstral speech description (Přibil & Přibilová 2008). The main advantage of this approach consists in a fact that only new cepstral description must be created and the original speech database is applied as a common area for all voices.

Statistical analysis of cepstral coefficients and the first three formant positions has shown that different emotional states are manifested in a speech signal in observed parameters. Spectrograms, histogram envelopes together with other parameters may be used for identification of individual emotions. This method can also be used for evaluation of emotional synthetic speech as a supplementary approach parallel to the listening tests.

From visual comparison of these spectrograms and histograms follows that emotional speech brings about the most significant spectral changes for voiced speech. Therefore, the extended analysis of sounds based on Welch’s periodograms was subsequently performed. Comparison of calculated spectral distances between “neutral” and transformed emotional styles of voiced sounds shows that the spectral changes (formant position and bandwidth) are the greatest for "angry" and the smallest for "joyous" style. These results are in correspondence with the applied emotional transformation method which means this approach is fully usable for detailed spectral analysis of voiced parts of speech. But a weak point of this method is the manual selection of ROIs. Speech recognition approach (Vích et al. 2008) can be used here (e.g. in the form of a simple phoneme alignment procedure) to get these ROIs automatically.

Results of the spectral flatness ranges and values statistical analysis show good correlation for both types of voices and all three emotions. The greatest mean SFM value is observed in "anger" style for both voices. Similar shape of SFM histograms can be seen in Fig. 19a) and Fig. 20a) comparing corresponding emotions for male and female voices. On the other hand, it was confirmed that only SFM values calculated from voiced frames of speech give sufficient information – in Fig. 21 it is evident that the histograms are practically the same for all three emotions. This subjective result is confirmed by the objective method – multiple comparison of groups based on results of ANOVA statistics and hypothesis test. Our final aim was to obtain the ratio of mean values which can be used to control the high frequency noise component in the mixed excitation during cepstral speech synthesis of voiced frames (Vích 2000). From summary results follows that the ratio of mean values is 1.18 times higher for female voice than for male voice.

From comparison of basic statistical microintonation analysis follows that absolute jitter values are in accordance with the human vocal tract properties. But there should be a problem with accuracy of jitter measurements, caused by the fact that jitter estimation on running speech (in contrast to steady vowels) is very difficult (Sun et al. 2009). Female shorter pitch periods are accompanied with shorter values of the absolute jitter but higher relative changes in the frequency domain (mean $F0_{DIFF}$ values). The highest values of jitter correspond to "joy" and the lowest ones correspond to "sadness" for both voices. Similar results are shown in (Tao et al. 2006). The same tendency can be observed for statistical results of zero crossing analysis. Although different frame lengths were used in microintonation frequency analysis for male and female voices, we can see matched similar values for all corresponding emotions. Visual comparison of histograms of zero crossing periods L_Z is not significant but higher relative occurrence of low L_Z values can be noticed in "neutral" style for both voices. On the other hand, as regards visual comparison of average spectra, similar curves can be matched in Fig. 26 and Fig. 27 for male and female voice for all corresponding emotions. Obtained results of microintonation spectral analysis (especially the B_3 values) can be used to synthesize a digital filter for suppression of microintonation component of a speech signal. From objective statistical comparison of zero crossing periods by Ansari-Bradley hypothesis test follows that the null hypotheses were rejected at the 5% significance level for each emotion type inside the gender group and simultaneously, the null hypotheses between the corresponding emotions of both types of voices are in all cases fulfilled at the same significance level. The result of final multiple comparison of ANOVA also confirms good correlation between particular emotions.

In the next future, we plan to use results of ANOVA and hypothesis test for creation of the database of values for emotional speech classifier based on statistical evaluation approach (Iriondo et al. 2009), or it can be used for identification of speaker emotional states or in real-time emotion recognition systems (Attasi & Smékal 2008).

5. Acknowledgment

The work has been done in the framework of the COST 2102 Action "Cross-Modal Analysis of Verbal and Non-Verbal Communication". It has also been supported by the Grant Agency of the Czech Republic (GA102/09/0989), by the Grant Agency of the Slovak Academy of Sciences (VEGA 2/0090/11) and the Ministry of Education of the Slovak Republic (VEGA 1/0903/11).

6. References

Atassi, H. & Smékal, Z. (2008). Real-Time Model for Automatic Vocal Emotion Recognition. In: *Proceedings of 31st International Conference on Telecommunications and Signal Processing*, Parádfürdö, Hungary, pp.21-25, 2008

Boersma, P. & Weenink, D. (2008). Praat: Doing Phonetics by Computer (Version 5.0.32) [Computer Program]. Retrieved August 12, 2008 from http://www.praat.org/ Everitt, B. S. (2006). *The Cambridge Dictionary of Statistics*. Third Edition. Cambridge University Press, 2006

Fant, G. (2004). *Speech Acoustics and Phonetics*. Kluwer Academic Publishers, Dordrecht Boston London, 2004

Farrús, M., Hernando, J. & Ejarque, P. (2007). Jitter and Shimmer Measurements for Speaker Recognition. In: *Proceedings of Interspeech 2007*, Antwerp, Belgium, pp. 778-781, 2007

Hartung, J., Makambi, H.K. & Arcac D. (2001). An extended ANOVA F-test with applications to the heterogeneity problem in meta-analysis. *Biometrical Journal*, 43(2), 135–146.

Hosseinzadeh, D. & Krishnan S. (2008). On the Use of Complementary Spectral Features for Speaker Recognition. *EURASIP Journal on Advances in Signal Processing,* Vol. 2008, Article ID 258184, 10 pages, doi:10.1155/2008/258144, Hindawi Publishing Corp.

Iriondo, I., Planet, S., Socoro, J.C, Martínez, E., Alías, F. & Monzo, C. (2009). Automatic Refinement of an Expressive Speech Corpus Assembling Subjective Perception and Automatic Classification. *Speech Communication* Vol. 51, pp. 744–758, Elsevier, 2008.

Kuwabara, H. & Sagisaka, Y. (1995). Acoustic characteristics of speaker individuality: Control and conversion. *Speech Communication* Vol. 16, pp. 165-173, Elsevier, 1995

Li, X., Tao, J., Johnson, M.T., Soltis, J., Savage, A., Kirsten M., Leong, K.M. & Newman, J.D. (2007). Stress and Emotion Classification Using Jitter and Shimmer Features. In: *Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASPP '07)*, Honolulu, HI, pp. IV-1081-IV-1084, 2007

Madlová, A. & Přibil, J. (2000). Comparison of Two Approaches to Speech Modelling Based on Cepstral Description. In: *Proceedings of the 15th Biennial International EURASIP Conference Biosignal 2000*, Brno, Czech Republic, pp. 83-85, June 21-23, 2000

Nwe, T. L., Foo, S. W. & De Silva, L. (2003). Speech Emotion Recognition Using Hidden Markov Models. *Speech Communication* Vol. 41, pp. 603–623, Elsevier, 2003

Oppenheim, A.V. Schafer, R.W. & Buck, J.R. (1999). *Discrete-Time Signal Processing*. Second Edition. Prentice Hall, 1999

Přibil, J. & Přibilová, A. (2008). Application of Expressive Speech in TTS System with Cepstral Description. In: Esposito, A., et al. (eds.) Verbal and Nonverbal Features of

Human-Human and Human-Machine Interactions: *Lecture Notes in Artificial Intelligence 5042*, pp. 201-213, Springer-Verlag: Berlin Heidelberg, 2008

Scherer, K.R. (2003). Vocal Communication of Emotion: A Review of Research Paradigms. *Speech Communication*, Vol. 40, pp. 227–256, Elsevier, 2003

Srinivasan, S. & DeLiang, W. (2010). Robust speech recognition by integrating speech separation and hypothesis testing. *Speech Communication*, Vol. 52, 72-81, Elsevier, 2010

Stevens, K.N. (1997). Models of speech production. In: Crocker, M.J. (Ed.), *Encyclopedia of Acoustics*, pp. 1565-1578, John Wiley & Sons, Inc. 1997

Suhov, Y. & Kelbert, M. (2005). *Probability and Statistics by Example. Volume* I. Basic Probability and Statistics. Cambridge University Press 2005

Sun, R., Moore, E. & Torres, J.F. (2009). Investigating Glottal Parameters for Differentiating Emotional Categories with Similar Prosodics. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2009*), Taipei, Taiwan, pp. 4509-4512, 2009

Tao, J., Kang, Y. & Li, A. (2006). Prosody Conversion from Neutral Speech to Emotional Speech. *IEEE Transactions on Audio, Speech, and Language Processing* Vol. 14, pp. 1145–1154

Vích, R. (2000). Cepstral Speech Model, Padé Approximation, Excitation, and Gain Matching in Cepstral Speech Synthesis. In: *Proceedings of the 15th Biennial EURASIP Conference Biosignal 2000*, Brno, Czech Republic, pp. 77–82, 2000

Vích, R., Nouza, J. & Vondra, M. (2008). Automatic speech recognition used for intelligibility assessment of text-to-speech systems. In: Esposito, A., et al. (eds.) Verbal and Nonverbal Features of Human-Human and Human-Machine Interactions: *Lecture Notes in Artificial Intelligence 5042*. pp. 136-148. Springer-Verlag: Berlin Heidelberg, 2008

Volaufova J. (2005). Statistical Methods in Biomedical Research and Measurement Science. *Measurement Science Review*, Vol. 5, No. 1, Section 1, pp. 1-10, Versita, 2005

10

Speech Interface Evaluation on Car Navigation System – Many Undesirable Utterances and Severe Noisy Speech –

Nobuo Hataoka[1], Yasunari Obuchi[2],
Teppei Nakano[3] and Tetsunori Kobayashi[3]
[1]Department of Electronics and Intelligent Systems,
Tohoku Institute of Technology,
[2]Central Research Laboratory, Hitachi Ltd.,
[3]Waseda University,
Japan

1. Introduction

Recently, ASR (Automatic Speech Recognition) functions have commercially been used for various consumer applications including car navigation systems. However, many technical and usability problems still exist before ASR applications are on real business use. Our goal is to make ASR technologies for a real business use. To do so, we first evaluate a car navigation interface which has ASR as an input method, and second evaluate an ASR module using real noisy in-car speech. For ASR applications, we envision mobile environments, e.g. mobile information service systems such as car navigation systems and cellular phones on which an embedded speech recognizer (Kokubo et. al., 2006) is running and which are connected to remote servers that support various information-seeking tasks.

Taking a look at commercially available car navigation systems, currently over 75% systems have ASR interfaces, however, there are very few drivers who have experiences to use the ASR interfaces. What is the problem? This is caused by the ASR usability problems.

In this chapter, we report two experimental evaluation results of ASR interface for mobile use, especially for car navigation applications. First, we evaluate the usability aspects of speech interface and second, we evaluate in-car noise speech problems to propose an effective method to cope with noisy speech. For the first evaluation, we use a prototype which has a promising speech interface called FlexibleShortcuts and Select&Voice produced by Waseda University (Nakano et. al., 2007). We found many undesirable OOV (Out-Of-Vocabulary) utterances which make the interface worse. From the second experiment to check car-noise problems, we propose an array microphone + Spectrum Subtraction (SS) technique to increase recognition accuracy.

2. Problems of ASR car use

2.1 System concept of network applications

As information technology expands into the mobile environment to provide ubiquitous communication, intelligent interfaces will be a key element to enable mobile access to networked information. For mobile information access, HMIs (Human Machine Interfaces) using speech might be the most important and essential as speech interfaces are more effective for small, portable devices. Mobile terminals such as cellular phones, PDAs (Personal Digital Assistants) and Hand-held PCs (Personal Computers) are connected to networks like the Internet to access information from web servers. For mobile information access, speech processing and image processing will be key technologies on intelligent mobile terminals.

Especially, Car Telematics refers to a new service concept where mobile terminals (e.g. car navigation systems, cellular phones) are used to connect to networked information services. Figure 1 illustrates a total service system concept, which consists of three parts, e.g. terminal/client, network, and center/server. For the terminals, sophisticated HMIs are required to handle various inquiries and to deliver information from the center using speech and image input/output.

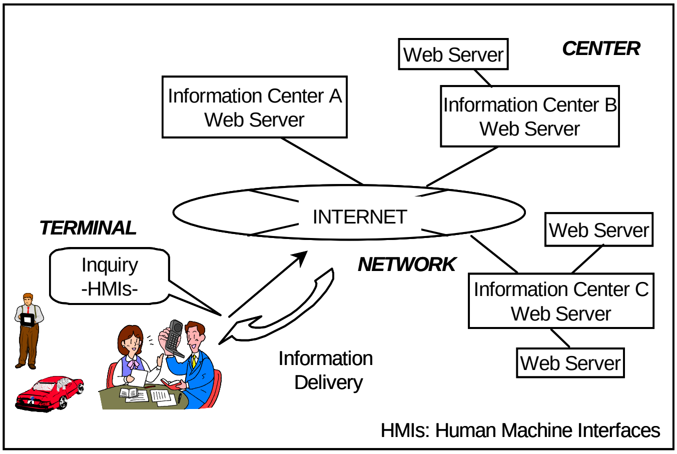

Web Server
CENTER
Information Center A Web Server
Information Center B Web Server
INTERNET
TERMINAL
Web Server
Inquiry -HMIs-
NETWORK
Information Center C Web Server
Information Delivery
Web Server
HMIs: Human Machine Interfaces

Fig. 1. Service System Image (Terminal, Network, and Center)

The network is typically the Internet; and via the Internet, the user's requests are transferred to related Web servers at the center, and required information will be provided from the center to users via networks and terminals (Hataoka et. al., 2004).

2.2 Technical t problems for automotive use

In cars, HMIs based on speech processing such as ASR and TTS are essential to provide a safe driving environment. However, there are many problems to be solved before a real use of ASR and TTS as follows;

1. **Usability problem:** All interfaces should have a transparent navigation model. However the interfaces using ASR do not have this function/feature. If the input is misrecognized, the user can not understand why the misrecognition occurred and then can not manage the next action.
2. **OOV (out of vocabulary) problem:** There are many ways to express one meaning/location. For example, we can say either “Starbucks” or “Staba” to show the same coffee shop. It is essential for the ASR to handle this OOV problem (Vertanen, 2008)

3. **Robustness issue:** The in-car speech has noise including an engine noise and audio noise etc. To enhance the degraded speech is essential for ASR. The array microphones are used to locate sound source and reduce environmental noise (Obuchi & Hataoka, 2006).

In this chapter, to deal with these problems, first we evaluated usability issues of ASR interface on a car navigation system, and second evaluated robustness of ASR module on the car navigation interface.

3. ASR interface evaluation

3.1 Pre-evaluation using commercial product

3.1.1 Evaluation setup and task

To make real problems of ASR interfaces clear, first, we evaluate a speech interface of a commercial product (Figure 2: PIONEER Carrozzeria AVIC-HRZ88II) using two environments. The first environment is in a laboratory room and the second one is in a noisy driving car. The number of subjects is 20 people. All are university students and no one has an experience of using speech interfaces. At the beginning, each subject was instructed how to use a car navigation ASR interface by an operator.

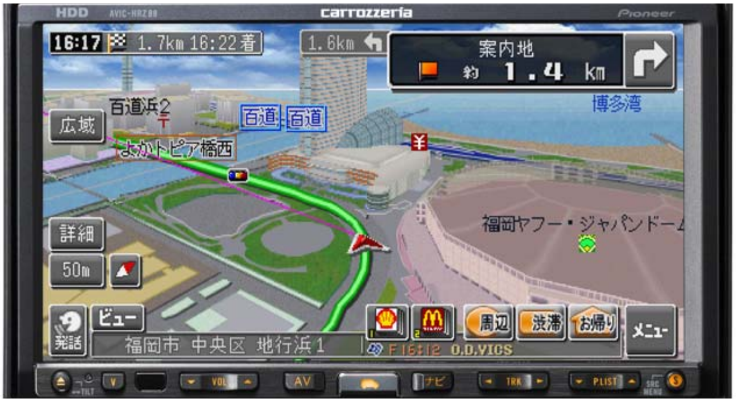


Fig. 2. Evaluated Car Navigation System (PIONEER)

Two tasks are evaluated under two environments of the room and the in-car. The evaluation tasks are as follows;

1. Command input: audio (radio/DVD etc) and air conditioner operation, i.e. "FM radio channel 4" etc.
2. Destination setting for navigation: Two utterance ways were evaluated, first utterance from the written vocabulary and second prompt utterance (free word)

3.1.2 Evaluation results

1. Room environment: Figure 3 shows evaluation results. The recognition success turn times are shown.
2. In-car environment: Figure 4 shows evaluation results of driving car environment. The results became worse than those of in room environment.

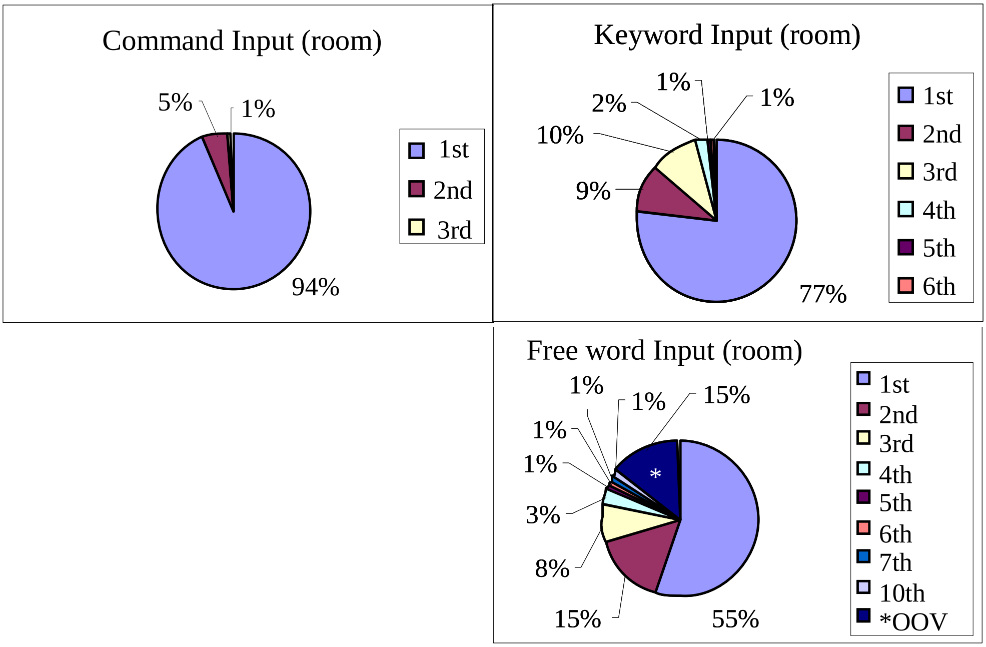


Fig. 3. Evaluation Results (laboratory room)

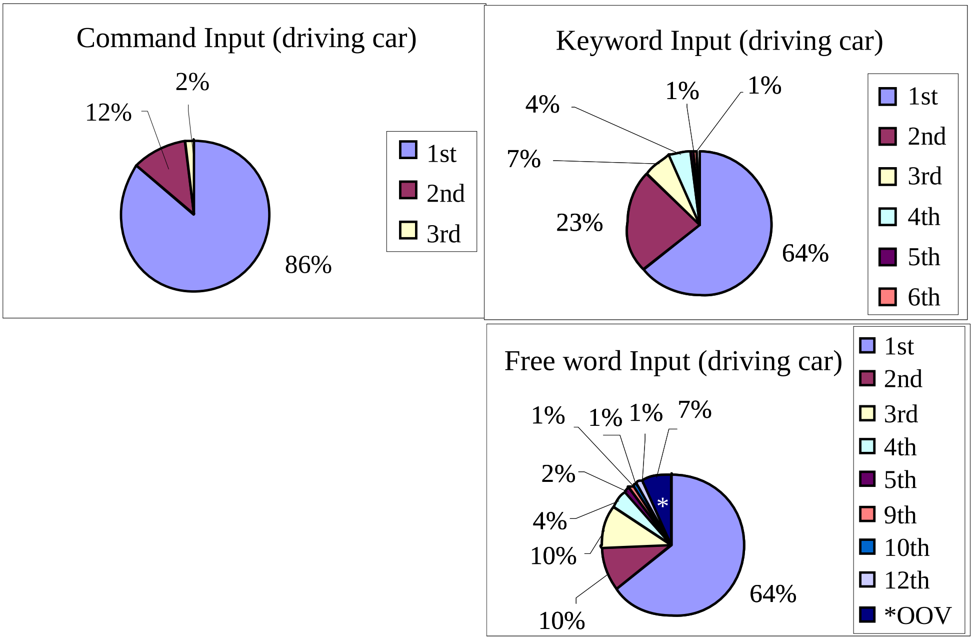


Fig. 4. Evaluation Results (driving car)

3.1.3 Consideration

The OOV issue was crucial which occurred at the free word utterance. Then we are developing an overcoming system to cope with this OOV problem. The system consists of terminals and centers, and when recognition errors and interruption of the speech input occur, the system sends all interaction records and speech data files to the center. At the center, a full specification of continuous ASR can recognize the data and then deliver a new vocabulary set to terminals.

3.2 Evaluation using prototype system

3.2.1 Philosophy for evaluation

First, we use the ASR prototype system called FlexibleShortcuts and Select&Voice. The FlexibleShortcuts can handle both of voice input and menu input and for the voice input many short-cuts are available to say related words directly. The Select&Voice has framed-based input windows to utter input words. The FlexibleShortcuts and Select&Voice are useful for car navigation task. Second, we use a location retrieval task in that many OOV utterances would be observed frequently in order to check how users act when the OOV utterances occur.

3.2.2 FlexibleShortcuts and Select&Voice

For the evaluation of OOV problems, we used a system consisting of FlexibleShortcuts and Select&Voice which have been developed by Waseda University (Nakano et. al., 2007). Waseda University is developing the Proxy-Agent as the platform of ASR application systems by the Japanese National Found Research Project called “Fundamental Technology Development on Speech Recognition.” This found was for three years from fiscal year of 2006 to 2008. The Proxy-Agent has characteristic features of plug-in based function merging and connection to network servers. The function merging is independent from ASR engines to do data collection, new ASR engine adding, and co-use of possible ASR parts etc.

The FlexibleShortcuts and the Select&Voice are the speech interface functions which are developed as an application development tool in the Proxy-Agent framework. The FlexibleShortcuts is a speech interface having flexible selection of speech inputs and/or menu inputs and also shortcut functions. In the menu expression, if a user knows the shortcut way meaning the final word to say, the user can utter this final word to reach the destination. For example, in the menu expression FM radio is under a radio category, but we can say “FM radio” directly to reach the FM radio handling process. The Select&Voice is a speech interface for data input which has been developed according to the analogy of GUI (Graphical User Interface). The Select&Voice has the framed-based input windows. This is named because the input processes are first “Select” input frame “And” then utter “Voice.”

The car navigation system which has the speech interface based on FlexibleShortcuts and Select&Voice is evaluated on the location/address retrieval task.

3.2.3 Evaluation experiment details

1. Evaluation Purpose

In this experiment, we evaluate whether the subjects can realize their OOV utterances and if so, when they can realize OOV utterances and how they act after realization using the framework of FlexibleShortcuts+ Select&Voice. We carefully select a task in which OOV will occur easily. From this viewpoint, we select a city and town setting task in which many city and town name changes occur after city and town merging by the local government. All actions are recorded as input and output logs using the Proxy-Agent architecture. Finally we analyze subjects' behaviors using recognition results and recorded logs.

2. Data Collection Environment

To focus on the OOV utterance and subject's action (behavior), the quiet environment of a laboratory room is set for the evaluation experiments.

3. Evaluated System/Equipment

We use the PC system consisting of FlexibleShortcuts+ Select&Voice developed by Waseda University on the car navigation task. Figure 5 shows a PC and a controller. Both menu selection and voice utterance are possible using the controller.

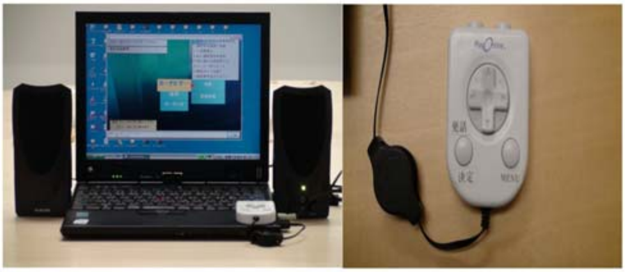


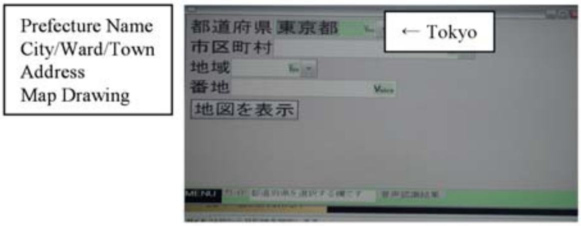
Fig. 5. PC and Controller used for Evaluation

4. Experimental Subjects

The number of subjects is 10 (ten) consisting of 5 (five) subjects who have experiences to use this kind of speech interface and 5 (five) subjects who have not experiences. The input and output logs and utterances are recorded using PC and a video recorder.

5. Utterance Conditions

Each subject retrieves ten (10) locations/addresses using speech utterance. In the ten locations, two names are changed by the new city merging. This means former names are OOVs. Figure 6 shows a display example of the Select&Voice interface showing prefecture, city, area, and address numbers.



都道府県 東京都
市区町村
地域
番地
地図を表示

Fig. 6. Display of Location Retrieval Task

3.2.4 Evaluation experiment setup

At first, an experimental operator introduces how to use the system to subjects and then subjects start evaluation experiments after short use of the system. In the evaluation stage, there is no advice and suggestion from the operator. The display of PC is recorded using a video recorder and all utterances and input and output logs are recorded in the PC memories.

3.2.5 Evaluation results

1. Evaluation Data for Location Retrieval

In the ten locations, two names of locations/addresses have been changed by the city merging. The following data show two names changed.

(1)before: Aijyo 1-1, Osato-machi, Osato-gun, Saitama
changed to Aijyo 1-1, Kumagaya-shi, Saitama
(2)before: Okisu 1-1, Kamisu-machi, Kashima-gun, Ibaragi
changed to Okisu 1-1, Kamisu-shi, Ibaragi

Others are three ordinance-designated city names which have area's name "ku." We normally do not understand how to separate "ku" and the following location names. Other 5 locations/addresses are normal names.

2. OOV Ratio

Table 1 shows evaluation results showing ratios of correct recognition, OOV, and misrecognition, respectively. In the experienced group, the total number of utterances is 352 and the number of correct recognition is 191 (54%), the number of OOV is 86 (24%), and the number of misrecognition is 74 (21%). In the non-experienced group, the total number of utterances is 601 and the numbers of correct recognition, OOV, and misrecognition are 304 (50%), 151 (26%) and 146 (24%), respectively.

Experienced			Non experienced	
	No.	Ratio	No.	Ratio
Correct recognition	191	54%	304	50%
OOV	86	24%	151	26%
Mis-recognition	74	21%	146	24%
bug	1		1	
total	352		601	

Table 1. Evaluation Results according to Experience

3. Feature of OOT Utterances

The varieties of OOV of the location/address retrieval task are complete OOV utterances, mis-division of ordinance-designated city names, input frame errors, and OOV utterances at the top display level of the FlexibleShortcuts stage. Table 2 shows details of OOV utterances. For the total 951 utterances, the number of OOV is 237 consisting of 112 complete OOVs, 55 address division errors, 21 input frame errors, and 49 top-display-level OOVs.

Complete OOVs	Address division errors	Input frame errors	Top display level	Total
112	55	21	49	237
47%	23%	9%	21%	

Table 2. The Number of OOV Utterances

4. Subject Action/Behavior after OOV Utterance

The following remarks are subjects' actions after OOV utterances.

a. Complete OOV utterances:

Table 3 shows an example in which a subject realized OOV after four (4) utterances (repeated). This subject realized an OOV utterance checking a vocabulary list shown using the controller. Figure 7 shows the number of utterances when subjects realized OOV utterances. The numbers of cases in which subjects realized OOVs at the second utterance, third one, forth one, fifth one, sixth one, seventh one, eighth one, and ninth one are 1 (5%), 3 (15%), 4 (20%), 1 (5%), 2 (10%), 6 (30%), 0 (0%), and 3 (15%), respectively.

Utterances	Recognition results	
Content	Recognized word	Reason
Ibaragi	Ibaragi	(correct)
Kashima-gun Kamisu-cho	Kusumigaura-shi	OOV
Kashima-gun Kamisu-cho	Sarushima-gun, Gosumi-cho	OOV
Kashima-gun Kamisu-cho	Sarushima-gun, Sakae-cho	OOV
Kashima-gun Kamisu-cho	Sarushima-gun, Sakae-cho	OOV
To the next address retrieval step		

Table 3. Example of OOV Utterances (address task)

b. Address Division Errors

To retrieve ordinance-designated cities, users should utter city name and “ku” name together, however most people utter city name and “ku” name separately. (Example: Sapporo-shi and Chuou-ku separately instead of Sapporo-shi Chuou-ku) Firstly, subjects utter these city and “ku” names separately, but when connected error output of city and “ku” names appeared on the display subjects understood how to utter ordinance-designated city names. Finally subjects could retrieve locations/addresses correctly. This is advantage of the Select&Voice architecture based on input frames.

c. Input Frame Errors

Subjects should select input frames correctly. If not so, subject's utterance will be an OOV utterance. From the check of logs, subjects can change the input frames correctly after realizing the misrecognition results in this case.

The improvement of ASR interfaces and how to deal with OOV utterances are the most important and urgent issues to be solved. To create new vocabularies (the same meaning, but different utterance styles) from original vocabulary automatically is one of future research issues.

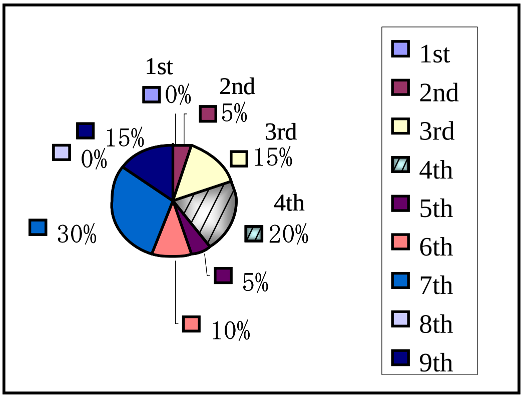


Fig. 7. The No. of Utterances when Subjects Realized OOV

4. In-car noise speech evaluation

4.1 Speech data collected in driving car

For the robustness of ASR interface, we evaluate in-car noisy speech and propose an effective pre-processing technique to cope with in-car noise. We used in-car speech data which were collected in moving cars using array microphones (Waseda 2007).

The recording was done in downtown Tokyo where the car was forced to drive slowly with frequent stops due to the traffic jam. Therefore, a large part of the background noise was from the surrounding environment such as other cars, constructions, etc. The speaker was sitting on the passenger seat, and there was a linear microphone array on the dashboard in front of the speaker shown Figure 8. The array consists of 7 (seven) microphones, which are located at the interval of 10cm, 5cm, 5cm, 5cm, 5cm, and 10cm. The array microphones were labeled as #1 to #7 from the driver seat side to the window side, so the central microphone was #4. Also, the headset microphone (#8) was used to collect noise-free speech.

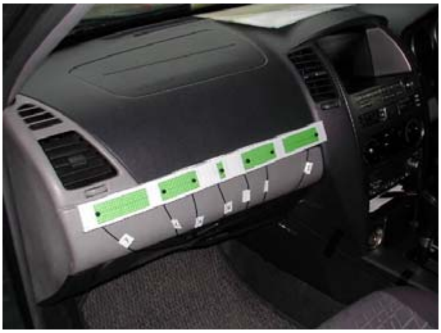


Fig. 8. A Microphone Array in Car

4.2 Original data analysis

We have collected the speech data from 18 speakers (11 males and 7 females, all in their early twenties). 3,620 utterances for 152 POIs (Point of Interests) were collected in total, and they were roughly segmented using a fixed time period from the beep. After segmentation, the length of the data was approximately 7 hours in total. The number of utterances per speaker ranged from 134 to 326, and the number of utterances per POI ranged from 10 to 48.

We then estimated the signal-to-noise ratio (SNR) by comparing the power of speech and non-speech segments. Table 4 shows the estimated SNR for each microphone. Since the noise spectrum has a strong peak in the low-frequency range, we also calculated the SNR after applying a bandpass filter of 400Hz to 5500Hz range. It is interesting that the estimated SNR does not have any correlations with distance between speakers and microphones, although speech recognition rate has correlations with distance of speakers and microphones. The recognition rate of headset microphone #8 is the best and would be the target rate.

Figure 9 shows ASR recognition rate for each microphone after low power cut processing. The low power cut frequencies are 20Hz for males and 40Hz for females.

mic. ID	SNR (full band) dB	SNR (400-5500Hz) dB	Rec. rate(%)
1	-0.5	9.3	83.2
2	-2.8	12.1	86.3
3	-3.4	8.6	86.8
4	-3.0	9.2	87.8
5	-2.7	11.7	88.7
6	-3.8	8.5	85.6
7	-2.9	10.5	76.4
close-talk	56.7	83.2	95.2

Table 4. Estimated SNR of Each Microphone Data (Obuchi & Hataoka, 2006)

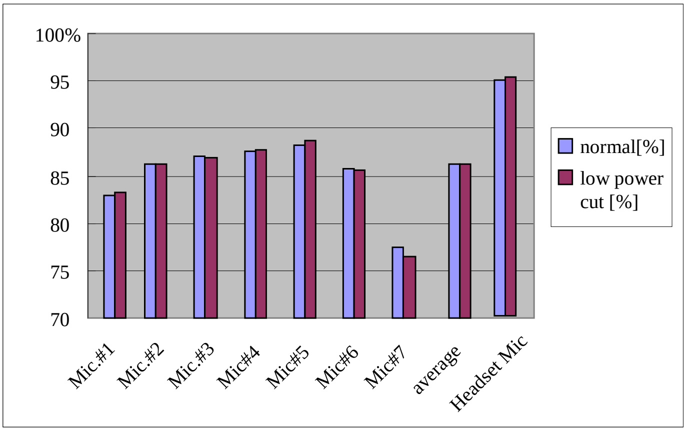


Fig. 9. Feature of Each Microphone

4.3 Experimental procedure

4.3.1 Free/open CSR software Julius/Julian

Free CSR (Continuous Speech Recognition) software Julian (Julius/Julian URL) was used as an ASR engine. There are two types of CRS engines such as Julian and Julius. The Julius is using language models based on N-gram and the Julian is using language models based on network grammars. The only difference between Julius and Julian is the language model. Both engines are using the same speech analysis and the same search algorithm. The search algorithm is based on two pass algorithm; the first search is a rough search using mono-phone HMMs and the second search by tri-phone HMMs. Julius is using bigram for the first search and trigram for the second search. We evaluate many noise handling techniques using a Julian decoder in automotive environments to make technical problems of speech processing modules clear.

4.3.2 Recognition experiments using various techniques

We carried out evaluation experiments of 152 POI isolated word recognition using Julian decoder for 5-male and 5-female data on a Linux machine. For Julian conditions, the sample acoustic model with PTM triphones was used. Among various variations of Julian, the Julian-v3.4.2 grammar driven decoder with 12 MFCC and log power, plus their first-order time derivatives is used. All the data were originally sampled by 44.1kHz, but down-sampled to 16kHz prior to the experiments.

The results of the baseline experiments (Table 1) showed the recognition rate of distant-talk was 88.7% (mic. #5) and that of close-talk was 95.2%. In the experiments, the individual recognition rate ranged from about 81% to 92% (average of all microphones).

The following pre-evaluation experiments are carried out to check problems of noisy speech data.

1. **Evaluation of Low Power Cut and Spectrum Subtraction (SS)**

According to many reports, the engine noise is ranging under 100Hz, so we carried out low power cut before the recognition stage. We checked frequency ranges of low power cut from 20Hz to 100Hz by 20Hz step size. Figure 10 shows results of low power cut (mic.#5). The cut of frequency 20Hz showed the best results (all average 88.8%), especially to female data.

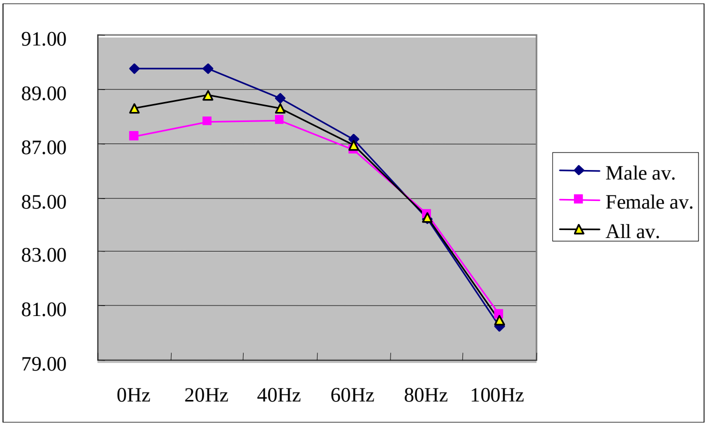


Fig. 10. Evaluation Experiments of Low Power Cut

There are many parameters for setting SS processing. We used standard function supported by the Julius/Julian software. Figure 11 shows all evaluation results, e.g. original data, low power cut data, and SS data (for all distant-talk microphones). These results show that the recognition rate and the pre-processing effects depend on talkers deeply.

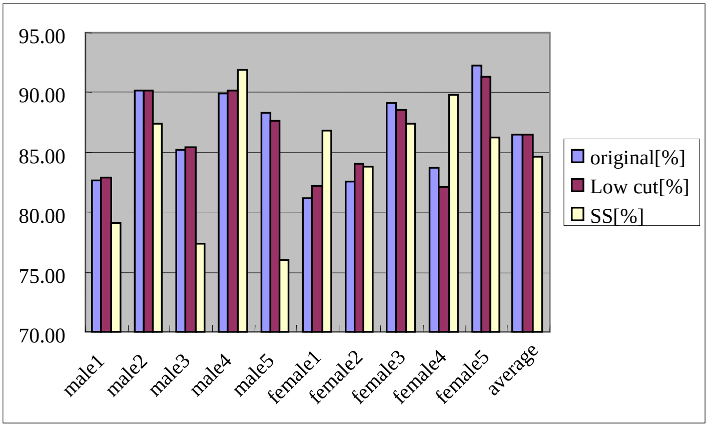


Fig. 11. Summary of Evaluation Experiments (Original/Low Power Cut/SS)

2. **Weighted Summation of Array Microphones (WS) + Spectrum Subtraction (SS)**

We tried the combination of array microphones and Spectrum Subtraction (SS). First, we summed speech data of possible array microphones, and then the SS processing was carried out to summed speech data. This array microphone technique is called Weighted Summation of Array Microphones (WS).

For SS, the following equation is used and two parameters *a*, *β* are checked from the viewpoint of recognition accuracy.

$$S(f) = X(f) - \alpha \cdot \hat{N}(f)$$
$$\hat{N}(f) = (1 - \beta)\,\hat{N}'(f) + \beta \cdot N(f) \qquad (1)$$

$\hat{N}(f)$: estimated noise, $\hat{N}'(f)$: previous estimated noise
α: alpha parameter, β: floor parameter

In the Julian software, the default of *a* = 2.0, and the default of *β* = 0.5. We checked *a* = 2.0, 3.0, 3.5, 4.0 and 5.0.

For WS, three types of summation of microphones are used; microphones #3 + #4, microphones #4 + #5, and microphones #3 + #4 + #5.

Figure 12 shows recognition rate using the SS pre-processing method according to the value of parameter *a*. The pair of microphone #4 and #5, and the parameter *a* = 3.5 gave the best recognition accuracy. For all pairs of #3, #4, and #5, the high average was obtained by *a*=4.0. However, the case of *a* = 3.5 gave a recognition dip for the pair of #3 and #4.

Figure 13 shows recognition results according to pre-processing, i.e. SS, WS, and SS+WS. The headset microphone gave the best recognition accuracy (around 98%), and WS of microphone #4 and #5 (+#3), and SS gave the second best accuracy (around 90%). However, there is still a big gap between the headset and array microphones + SS pre-processing.

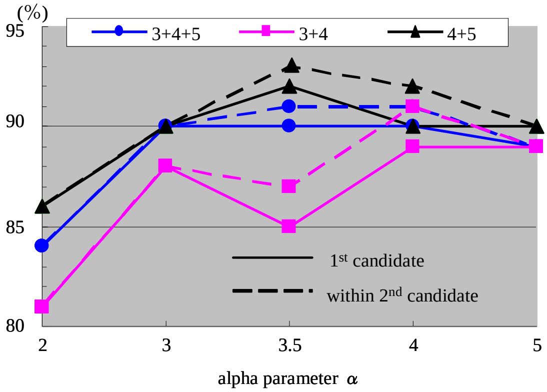


Fig. 12. Recognition Rate according to Alpha Parameter α

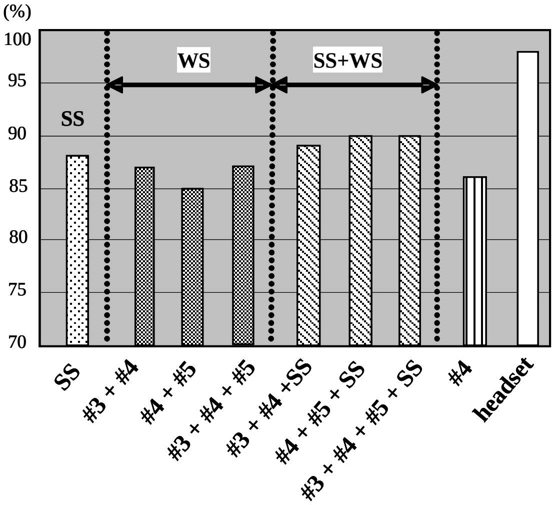


Fig. 13. Recognition Results according to Pre-processing

4.3.3 Consideration

In this work, we carried out possible tecuniques of the signal processing level to get a robust noise reduction method. Especially, we tried Weighted Summation of Array Microphones and Spectrum Subtraction. We obtained improvement using WS + SS, however there is still a big gap of recognition rate between the headset (98%) and WS + SS (90%), i.e. 8%. The recognition rate by the headset is the target one, so the further analysis is needed to reach the target accuracy by the pre-processing techniques.

5. Future work

The improvement of ASR interfaces and how to deal with OOV utterances are the most important and urgent issues to be solved. Also, more compact and more noise robust embedded version of Julius should be developed. For the noise robust technique, we are trying the subtractive array-microphone method for the adaptive noise estimation. Moreover the use of Missing Feature Theory (Raj et. al., 2005) is a promissing one for the noisy speech.

6. Conclusions

This chapter described two experimental evaluation results of ASR interface for mobile use, especially for car navigation applications. First, we evaluated the usability aspects of speech interface on a car navigation system and second, we evaluated in-car noisy speech by the various pre-processing techniques. For the first evaluation, we used a prototype which has a promising speech interface called FlexibleShortcuts and Select&Voice produced by Waseda University. To check OOV (Out-Of-Vocabulary) problems, we used the special case in that many location names have been changed by the town/city merging. This means that previous old location names are OOVs. For the location setting applications on the car navigation system, we found many undesirable OOV utterances occurred which made the interface worse. From the second experiment to check car-noise problems, we propose the combination method of Weighted Summation of Array Microphone (WS) + Spectrum Subtraction (SS) to increase recognition accuracy.

7. Acknowledgement

This research work was supported by the fund of “Fundamental Technology Development on Speech Recognition” organized by Japanese Ministry of Economy, Trade and Industry (METI). The speech data collected in the driving cars were obtained from Central Research Laboratory, Hitachi ltd. by the research project supported by New Energy and Industrial Technology Development Organization (NEDO) in Japan.

8. References

Kokubo, H, et al. (Oct. 2006). Embedded Julius: Continuous Speech Recognition Software for Microprocessor, MMSP2006, Canada

Nakano, T, Fujii, S, and Kobayashi, K. (Dec. 2007). Extensible Speech Recognition System using Proxy-Agent, Proc. of ASRU2007, pp. 601- 606

Hataoka, H, et al. (Jan. 2004). Robust Speech Dialog Interface for Car Telematics Service, Proc of IEEE CCNC2004, Las Vegas

Vertanen, K. (Apr. 2008). Combining Open Vocabulary Recognition and Word Confusion Networks, ICASSP2008, SPE-P4, G7, Las Vegas

Obuchi, Y. and Hataoka, N. (Sept. 2006). Development and Evaluation of Speech Database in Automotive Environments for Practical Speech Recognition Systems, Proc. of Interspeech2006, Pittsburgh, PA, USA

Waseda University IT Institute. (2007). Advanced Research on Speech Recognition Technologies, C-3 to C-52

Julius/Julian URL – an Open Source Large Vocabulary CSR Engine-:
http://julius.sourceforge.jp/

Raj, B. and Stern, R. M. (Sept. 2005). Missing-Feature Approaches in Speech Recognition, IEEE Signal Processing Magazine, Vol.22, No. 5, pp.101-116

Part 5

Speaker Diarization

11

A Review of Recent Advances in Speaker Diarization with Bayesian Methods

Themos Stafylakis[1] and Vassilis Katsouros[2]

[1]*Institute for Language and Speech Processing, "Athena"R.C. & National Technical University of Athens, Department of Electrical and Electronic Engineering*
[2]*Institute for Language and Speech Processing, "Athena" R.C.*
Greece

1. Introduction

This chapter aims to present some of the recent Bayesian approaches to speaker diarization (SD). SD is the task of grouping an audio document into homogenous regions, where each region should ideally correspond to the complete set of utterances that belong to a single speaker. Rich transcription, speaker adaptation of speech recognition systems and speaker recognition are some of the applications that require such a clustering procedure. Broadcast News, meeting, and telephone conversations are the main domains that SD is applied to.

SD is a fully unsupervised clustering task. Not only we are not allowed to use any target-speaker enrollment data to detect the target speakers through the acoustic stream, but the number of speakers should be considered as an unknown, too. Moreover, text-independence should also be assumed, meaning that no transcript is available, either.

Despite the effectiveness of several approaches and frameworks that have been proposed and tested in literature, the most natural and systematic approach to SD is to treat it as a model's order selection task. Once the order is estimated (i.e. the number of speakers) the task reduces to a familiar (but not trivial at all) machine learning task where the latent variables (i.e. the speaker indicators of each utterance) of given cardinality should be estimated from the observations. Therefore, a major issue we deal with is how to assess the number of speakers in a way that is simultaneously robust and efficient.

Bayesian machine learning is a highly principled paradigm and can naturally tackle model selection problems. It does so by applying consistently the rules of probability in order to infer the desired quantities, including the order of the model. Its superiority over the frequentistic statistical framework (e.g. Maximum Likelihood estimates, Classical Hypothesis testing) or semi-Bayesian approaches (e.g. MAP estimation, penalized maximum likelihood criteria) in model selection, averaging and density estimation has been verified in most (if not all) of the speaker related tasks, including identification and verification.

Several drawbacks however still exist, most of which stem from the intractability of the majority of the ideal Bayesian solutions. Many well known and effective machine learning tools cannot be applied or require severe adaptation that may drastically increase their computational complexity. Nevertheless, the introduction of powerful approximate inference method (e.g. Variational Bayes, Expectation Propagation), novel Markov-Chain Monte Carlo techniques, along with the rapid development of the Bayesian nonparametric models

(Infinite-HMMs, Dirichlet process mixture models, a.o.) allows us to create new approaches that are based on the statistical coherency of the Bayesian framework.

The rest of the chapter is organized as follows. In Section 2, some of the non- and semi-Bayesian approaches to SD is reviewed, along with some definitions and general algorithms strategies. In Section 3, the basic theory of Variational Bayes inference is presented with emphasis on mixture models, while a Variational Bayes algorithm that uses supervectors is examined in Section 4. In Section 5, we consider the use of infinite models to SD, while some ideas about further applications of Bayesian inference in diarization are discussed in Section 6. Finally, an introduction to some novel features that are utilized in speaker verification and recently in diarization are presented in the Appendix.

2. Short overview of speaker diarization approaches

In this section, a brief introduction to SD is presented, followed by some approaches that have been proposed in literature. We will refer to several algorithmic approaches and discuss some of their strengths and weaknesses. For a more complete overview of these methods we refer to (Tranter & Reynolds, 2006).

2.1 Front-end features and preprocessing steps

Before we examine the several algorithmic approaches, let us review some aspects that are common to all systems. The majority of SD systems use Mel-Frequency Cepstral Coefficients (MFCC) as front-end features, although other feature spaces have been proposed, such as Linear Frequency Cepstral Coefficients (LFCC) and Perceptual Linear Predictive (PLP), (Hermansky et al., 1985). Some systems utilize prosodic features to augment the cepstral representation (see Friedland et al. (2009)) while other approaches attempt to fuse several spaces and increase the diarization accuracy, (Gupta et al., 2007). Depending on the application field, one may consider techniques to normalize the MFCC stream, (Pelecanos & Sridharan, 2001), (Xiang et al., 2002), (Hermansky et al., 1992). These techniques aim to remove the linear channel effect and possibly the additive noise introduced by the recording chain, and are compulsory when a speaker may speak with more than one recording chains. In SD, such techniques may not be necessary; a standard assumption is that each speaker speaks only under identical conditions, i.e. recording equipment and background noise. Moreover, since the channel is unknown, these techniques unavoidably remove information that is related to the speaker and therefore increase the similarity between different speakers.

In the multiple-microphone setting (e.g. meetings), two are the main approaches. The first is to apply acoustic array processing techniques (i.e. beamforming) in order to mix the signals into a unique enhanced signal, (Anguera et al., 2007). A second approach is to utilize the estimated direction-of-arrivals (DOA) and fuse spatial and cepstral information, (Pardo et al., 2007). In our review, we will focus on the former approach when multiple microphones are in-hand.

A second step that is common to most of the algorithms is Speech Activity Detection (SAD). Silent regions of duration more that 200*ms* should be detected and removed from the steam. The official scoring method of NIST, the Diarization Error Rate (DER), penalizes false alarm and missed detection rates linearly. A common approach to detect speech is to assume that speech and silence follow a normal distribution each, in the log-energy domain. An Expectation Maximization (EM) algorithm with two Gaussian components is then applied, using the log energy as features. The energy feature stream is calculated using sliding windows of typically 30*ms* duration, with 20*ms* overlap, so that it is aligned with the MFCC stream. Temporal smoothing techniques are then applied on the binary labels to discard regions of less that 200*ms* duration. Hidden Markov Model (HMM)-based EM may also be considered as well, in order to avoid the need of ad-hoc or morphological filtering techniques. Apart from the energy, periodicity based methods have been proposed. These methods utilize the facts that vowels exhibit strong (quasi-)periodicity and apply it to discriminate speech from silence. Periodicity based approaches are usually more robust to noisy environments, however they require more computational effort than the energy-based ones, (Ishizuka et al., 2010).

Finally, in the Broadcast News field, most systems discriminate between acoustic classes like speech, music, music and speech, and silence. To do so, supervised learning techniques are applied. Each class is modeled with a GMM with 128 or 256 diagonal Gaussian components using labeled training data. During the classification stage, regions that are classified as non-speech are removed from the stream, after a proper temporal smoothing on the class-label domain.

2.2 General algorithmic approaches

After the preprocessing steps described above, SD algorithms diverge into two main directions. Those that apply segmentation to the MFCC stream, which might be uniform or based on the speaker change detection algorithms (see (Chen & Gopalakrishnam, 1998)), and those that do not apply such segmentation. Following the terminology of (Meignier et al., 2006) we will refer to the former branch as step-by-step algorithms, while to the latter as integrated algorithms. Both algorithmic approaches exploit a certain characteristic that the speaker labels exhibit, which is the temporal continuity. To realize the minimum range of this continuity, note that a speaker's turn lasts no less than 1 or 2*s* while the MFCC rate is 10*ms*, typically. Step-by-step algorithms exploit this continuity in order to turn the problem into a typical unsupervised clustering task. They represent each segments using a statistical model (a single Gaussian or a GMM) and they apply clustering techniques to group them into speakers. On the contrary, the integrated algorithms exploit the temporal continuity by assuming that the transitions between speakers follow a stochastic process which can be modeled by a (first-order and time-independent) Markov chain, where the probability of self-transition is significantly greater than the one of departing from the current state. Since the labels are not directly observed (in fact, they are the desired quantities) an observation model should be added, to link each distinct label (or state) with the observations. The overall model is therefore a HMM, where the observation model (i.e. the state-emission probabilities) is usually a GMM for each state, that is capable of capturing the multimodality of the state-conditional distribution.

2.3 Distance-based and model-based approaches to speaker clustering

However, what restrain us from using standard clustering or HMM techniques is the lack of knowledge regarding the number of speakers, say K. If we a priori knew K then we would apply an EM-algorithm to learn both the model and the latent variables (i.e. the label of each MFCC frame). [1] Two are the main approaches to deal with this issue. The first approach, which is extremely common to step-by-step algorithms, is to apply agglomerative hierarchical clustering (AHC) to merge those segments being close enough, in a statistical sense. What is required is a measure of similarity (or equivalently dissimilarity) and usually a predefined

[1] This is partially true however; phoneme rate, pitch, intensity and other emotional variations that speakers may exhibit during their speech may cause failure even in this setting.

threshold. We refer to these approaches as distance-based. Most step-by-step approaches use a two stage AHC procedure; at the first stage the segments (and consequently the clusters) are modeled using a single Gaussian of full covariance matrix, while GMMs are deployed only on the next stage, to merge those clusters that had not been merged during the first step. Several of the similarity measures that are used in the second stage are discussed in the Appendix, along with the MAP-EM algorithm that is applied to train the GMMs. Note also that several hybrid algorithms exist as well. For example, the highly robust and tuning-free approach proposed in (Ajmera & Wooters, 2003) uses a uniform segmentation stage and applies a Viterbi re-segmentation algorithm each time a pair of clusters is merged. Finally, several other alternative to AHC algorithms have been proposed, including Self-Organized Maps, Spectral Clustering and the Mean-Shift algorithm, that produce competing or better SD results, (Lapidot, 2003), (Ning et al., 2006), (Stafylakis et al., 2010b).

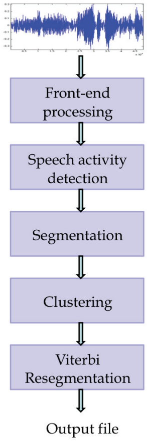


Fig. 1. Flow-chart of a baseline step-by-step algorithm

The main problem, however, regarding the distance-based approaches is their heuristic nature, in the sense that they do not propose a method to score overall clustering hypotheses. Note that the distance-based category of approaches may include even methods that rely on similarity measures that are derived from model-selection. For example, the local-Bayesian Information Criterion (BIC) ((Zhu et al., 2005), (Barras et al., 2004)) might be a model-based dissimilarity measure, however it does not correspond to the difference between scores of competing clustering hypotheses, (Stafylakis et al., 2010a). A desired property of a clustering algorithm is to be capable of providing a score to every single possible configuration of the latent variables. This is the essence behind the model-based approaches, (Fraley & Raftery, 1998). A model-based approach may be applied to a broad range of algorithms, which includes the AHC as well. To do so, we need to consider the dissimilarity between any pair of segments (or clusters of segments) as the increase or decrease of the overall score caused by the action of merging this pair. The global and the segmental settings of BIC are such examples, (Stafylakis et al., 2010a).

2.4 Penalized likelihood and its limitations

The most significant gain, however, from using model-based approaches is that it allows us to make use of the most natural and powerful tool of learning with missing data, that is the broad class of EM algorithms, (Dempster et al., 1977), (Amari, 1995). The integrated approaches typically use an evolving HMM (E-HMM), that is an HMM with increasing number of states. For each number of states, the Viterbi of Baum-Welch algorithm is deployed to learn the latent variables and estimate the emissions. To estimate the true number of speakers and the corresponding clustering hypothesis, an appropriate model selection method is compulsory. In the step-by-step approach, a form of EM algorithm can be applied instead of the AHC algorithm, over the range of a priori plausible number of speaker and a model selection method is required in order to select amongst them, (Mackay, 2003).

The penalized likelihood criteria have become very popular, for two main reasons. First of all, they can be used to apply model selection without altering the non- or semi-Bayesian way we estimate the parameters of the model with missing data. For example, in the E-HMM approach to SD, one may use the standard Maximum Likelihood (or MAP estimate) and penalize it according to the well-known BIC penalty term. The second reason is that under some regularity conditions they are limits of the desired Bayesian quantity; the *marginal likelihood* of the model. The BIC is derived by approximating the *marginal likelihood* of the model with the Laplace method, and discarding those terms that do not scale with the number of observations.

However, there are certain drawbacks regarding this semi-Bayesian approaches. For example, there are several models for with the consistency of the BIC has not been proven. This includes all the mixture models, including GMMs and HMMs as well. Even though in cases where the regularity conditions hold, the Laplace approximation is usually inaccurate for small sample sizes. Moreover, a MAP estimate is still point estimate, since the uncertainty about the estimate is being ignored, (Mackay, 2003). Finally, many of the powerful Bayesian tools, like the use of explicit priors or the use of hierarchies to tie several parameters cannot be combined in a profound way with the BIC approximation. Therefore, it becomes evident that a fully-Bayesian treatment of SD is required, which is the objective of the rest of this chapter.

3. Methods based on Variational Bayes approximate inference

In this chapter, the use of a fully Bayesian framework to SD is examined. The term Variational Bayes (VB) refers to a set of methods (the most popular of which being the *mean-field* VB) that approximate the desired quantities (e.g. marginal likelihoods, posterior probabilities, predictive densities) by bounding the marginal likelihood of the model from below. The use of VB is SD has been pioneered by F. Valente (Valente, 2005) and has been refined by P. Kenny *et al.* (Kenny et al., 2010) by applying it to i-vectors. We should emphasize that VB is a *general purpose* (approximate) inference method and its use is not limited to finite mixture models. On the contrary, it can be applied to nonparametric models, too (e.g. Dirichlet Process Mixture Models, (Blei & Jordan, 2005)).

3.1 Fundamentals of Variational Bayes

Let as consider a family of nested models $\mathcal{M}$ and let K denote the order of the model (e.g. the number of components of a GMM, the number of states of an HMM, etc.). Let the parameter space be denoted by Θ while the set of latent variables by X. The most probable order of the model given Y is the one that maximizes $P(K|Y) \propto p(Y|K)P(K)$. Assuming uniform prior over the hypothesis space (i.e. $P(K) \propto 1$), we need to maximize the *marginal likelihood* of the model with respect to (w.r.t.) K, i.e.

$$p(Y|K) = \int p(Y, X, \Theta) dX d\Theta \tag{1}$$

Alike BIC and other Laplace approximation based approaches, the VB framework defines a lower bound of (1). It does so by (i) introducing the variational posterior $q(X, \Theta)$ (the conditioning on Y is kept implicit) and (ii) applying the Jensen inequality, as follows

$$\log p(Y|K) = \log \int \frac{p(Y, X, \Theta)}{q(X, \Theta)} q(X, \Theta) dX d\Theta \geq \int \log \left(\frac{p(Y, X, \Theta)}{q(X, \Theta)} \right) q(X, \Theta) dX d\Theta \quad (2)$$

The bound that (2) defines is known as the (negative) Variational free energy $\mathcal{F}_K(q(X, \Theta))$, while the difference between $\log p(Y|K)$ and $\mathcal{F}_K(\cdot)$ is equal to $D_{KL}(q(X, \Theta)||p(X, \Theta|Y))$. However, no further improvement can be attained without making some assumptions about the functional form of $q(X, \Theta)$. The mean field VB pretends that $X|Y$ and $\Theta|Y$ are independent, and therefore assumes that $q(X, \Theta)$ admits a factorization of the form $q(X, \Theta) = q(X)q(\Theta)$. We say so, since this factorization is only a priori possible. A posteriori, the observation of Y induces an (at least weak) correlation between X and Θ. However, this independence assumption allows as to make the optimization problem tractable by applying calculus of variations.

3.2 The VB-EM algorithm

By maximizing $\mathcal{F}_K(q(X)q(\Theta))$ w.r.t. to $q(X)$ and $q(\Theta)$ we end up with the VB-EM algorithm described below

$$\text{VB-E step: } q(X) = \frac{1}{Z_X} e^{<\log p(Y, X|\Theta)>_{q(\Theta)}} \quad (3)$$

$$\text{VB-M step: } q(\Theta) = \frac{1}{Z_\Theta} e^{<\log p(Y, X|\Theta)>_{q(X)}} p(\Theta|K) \quad (4)$$

where $< a >_b$ denotes the expected value of a w.r.t b, while Z_X and Z_Θ are the corresponding normalizing constants. Note that the existence of $p(\Theta|m)$ at the M-step induces no asymmetry between X and Θ; the prior of X is incorporated through the complete-data likelihood $p(Y, X|\Theta) = p(Y|X, \Theta)p(X|m)$.

The severe distinction between ML-EM (or MAP-EM) and VB-EM is that while the former proceeds with simple point masses $\delta(\Theta, \hat{\Theta})$ placed at the ML or MAP estimates of Θ, VB-EM captures the uncertainty in these estimates, through the posterior distribution of Θ. Each estimate of X is obtained by averaging w.r.t. to the posterior of Θ, and not by $\delta(\Theta, \hat{\Theta})$. Furthermore, the benefits from using such a fully probabilistic approach are not restricted to obtaining much richer inferences about Θ and X. Contrary to ML- and MAP-EM, VB-EM aims to maximize the marginal likelihood of models, which is the key quantity in assessing K. No penalty term is required; the marginal likelihood is all we need to obtain in order to select between the rival models.

However, we should re-emphasize that the quantity being maximized by VB is $\mathcal{F}_K(q(X, \Theta))$ and not $\log p(Y|K)$. We saw that the difference between the two terms is equal to $D_{KL}(q(X|K)q(\Theta|K)||p(X, \Theta|Y, K)) > 0$ which increases with K. Therefore, the approximation of $\log p(Y|K)$ by $\mathcal{F}_K(q(X, \Theta))$ induces a systematic bias towards simpler models and therefore VB may underestimate the true number of speakers.

3.3 Hyperparameters: centering and strength

So far, we have assumed that the hyperparameters (i.e. the variables that parametrize the prior) remain fixed during the VB-EM. Let us denote the set of hyperparameters by H. By restricting ourselves to the *conjugate* family of priors, the hyperparameters can be distinguished into two sets $H = [H^c, H^s]$. Those that parametrize *expected values* of elements of

Θ (the prior centers, denoted by H^c) and those that determine the *amount of virtual observations* carried into the prior, also known as the *strength* of the prior (e.g. the relevance factor r in (42)), denoted by H^s. Priors with large strength are called *informative*, in the sense that their impact on the posterior is significant, at least when dealing with small or medium T. In cases where only vague, unreliable, or no information at all is available about Θ, a good strategy is to keep the prior as *non-informative* as possible. Jeffreys' priors, defined as follows

$$p^J(\theta) \propto |\mathcal{I}_\Theta(\theta)|^{1/2} \tag{5}$$

where $\mathcal{I}_\Theta(\theta)$ denotes the Fisher information matrix and θ an element of Θ, are flat in the sense that they place equal probability mass on each *natural volume element* of the statistical manifold, (Snoussi, 2005). They are also limits of conjugate priors, defined as below

$$p(\theta|h^c_\theta, h^s_\theta) \propto |\mathcal{I}_\Theta(\theta)|^{1/2} \exp\left(-h^s_\theta D_{KL}(h^c_\theta||\theta)\right) \tag{6}$$

by letting the strength go to zero. However, they are rarely *proper*, since $\int |\mathcal{I}_\Theta(\theta)|^{1/2} d\theta$ usually goes to infinity, and therefore inadequate for the model selection task. Hence, one may use conjugate priors and place h^s_θ equal to the minimum value (or the minimum integer) for which (6) is proper.

A further issue regarding the strength of the prior is whether the overall amount of virtual observations should remain fixed or be allowed to vary with K. For example, the standard penalty term of BIC implies a strength that remains fixed. Hence, the more parameters we add to the model, the less informative the (implied) prior will be for each single parameter. However, this strategy can be restrictive for models having parameters whose prior requires a minimum amount of strength to be proper (e.g. covariance matrices). In such cases, this strategy bounds from above the overall number of parameters that can be used and, consequently, the number of clusters. On the contrary, letting the strength grow with the number of parameters can cause overestimation of the true order of the model.

In any case, if we choose to optimize the hyperparameters, a straightforward solution is to solve the following maximization problem

$$H^{(i+1)} = \arg\max_{H} \mathcal{F}_K(q(X)q(\Theta), H^{(i)}) \tag{7}$$

As an alternative, hierarchical priors may be considered. In this approach, one may attach priors to the hyperparameters as well, that are governed by hyper-hyperparameters, and so on. Thus, one may consider marginalizing w.r.t to the parameters instead of maximizing. This approach is used in (Kenny, 2010) where a vague Gamma (hyper)-prior is attached to the precision of the Gaussian prior, resulting in an overall student-t prior distribution of the speaker factor. The experimental results of the 2010 speaker verification competition of NIST showed that the inclusion of this additional level of hierarchy increases significantly the verification accuracy.

4. A Variational Bayes approach to speaker diarization using supervectors

In this section, we examine in detail a VB approach to SD that utilizes supervectors in order to represent speech segments. Supervectors are high-dimensional vectors that are formed by concatenating the mean values of a GMM. The GMM is MAP-adapted from a Universal Background Model (UBM), where only the means are allowed to be adapted (see Appendix). Each supervector is then projected onto a space of lower dimensionality and VB inference in adopted to estimate the number of speakers and the assignment of segments to speakers. VB methods that do not make use of the supervector representation can be found in (Valente, 2005),

4.1 Supervectors and modeling assumptions

As explained in the Appendix, supervectors are high-dimensional vectors that are capable of capturing speaker characteristics in great detail, and are applied to speaker verification and recently in diarization, too. A main assumption used in the proposed method is that a supervector M can be described by a mid-dimensional vector w, as follows

$$M \approx M_0 + Vw \tag{8}$$

where M_0 the center of the acoustic space (i.e. the supervector of the UBM) and V a low rank (say p) rectangular matrix. The columns of V are the eigenvectors and have been extracted off-line. Furthermore, the columns of V are properly scaled with the corresponding eigenvalues so that $w \sim \mathcal{N}(0, I_p)$. Finally, let Σ be the diagonal covariance matrix of M_0 (see (Kenny et al., 2005) for a detailed derivation).

Let us assume (i) that a segmentation of the stream Y into segments has been applied. A uniform segmentation of 1s duration is proposed in (Kenny et al., 2005), however, speaker change detection techniques may be applied as well. The segmented MFCC stream is denoted by $Y = \{y_m\}_{m=1}^{M}$. For a given number of speaker K, an K-dimensional indicator vector i_m is used to indicate the speaker it belongs to, that is $i_{mk} = 1$, if and only if y_m belongs to the kth speaker. The collection of these vectors is denoted by $\mathcal{I} = \{i_m\}_{m=1}^{M}$. Moreover, the parameter vector of the kth speaker is denoted by w_k and their collection as $W = \{w_k\}_{k=1}^{K}$.

We further assume (ii) that an upper bound of the number of speakers (say K_{max}) is given and that the mixing coefficients $\pi = \{\pi_k\}_{k=1}^{K}$ (i.e. the prior probabilities of each speaker) can be estimated by maximizing the marginal likelihood

$$\int p(Y, W, \mathcal{I}|\pi) dW d\mathcal{I} \tag{9}$$

w.r.t. π. This technique, known as Maximum Likelihood II (ML-II) clearly diverges from the Bayesian framework. A fully-Bayesian approach attaches priors (e.g. Dirichlet) to $\{\pi_k\}_{k=1}^{K}$ and integrates out these parameters, too, instead of maximizing w.r.t. them. However, this technique enables us to estimate K without resorting to comparison between the marginal likelihood of several K, which can be time consuming when dealing with a large range of candidate number of speakers. On the contrary, by using this technique, we can estimate K simply by counting the number of mixture coefficients assigned non-zero values by ML-II, (Corduneanu & Bishop, 2001).

Finally, we assume (iii) that the alignment of frames with GMM-level mixture components is given. This assumption uses the final E-step of the EM algorithm as an estimate of the missing data (i.e. the component indicators). Using this assumption, we not only have to deal with the a single set of missing data, i.e. $\mathcal{I} = \{i_m\}_{m=1}^{M}$, but we are able to represent segments with *sufficient statistics* and utilize closed-form expressions to calculate the desired statistical quantities. This is due to the fact that the complete-data likelihood of a GMM belong to an exponential family, while the incomplete-data likelihood does not.

4.2 Working with the complete-data

To stress the benefits from the third assumption, let us derive some useful formulae that will be used, namely the likelihood, the posterior and the marginal likelihood of a *single* GMM that is represented by w. Let $y_u = \{y^t\}_{t=1,2,...}$ be the MFCC coefficients of a segments. We parametrize the (centralized) statistics of each segment as

$$N_c = \sum_t \gamma^t(c) \tag{10}$$

$$\tilde{F}_c = \sum_t \gamma^t(c) \left(y^t - \mu_c^0\right) \tag{11}$$

and

$$\tilde{S}_c = diag\left(\sum_t \gamma^t(c) \left(y^t - \mu_c^0\right)\left(y^t - \mu_c^0\right)^T\right) \tag{12}$$

where $\gamma^t(c)$ the posterior probability that y^t belongs to the cth component, given by the MAP-EM algorithm. This is our estimate of the missing data, that is already in-hand from the MAP-EM algorithm. For notational compactness, let us define $\mathbf{N}$ the $Cd \times Cd$ diagonal matrix, whose C diagonal block are defined as $\{N_c I_d\}_{c=1}^C$. Let also $\tilde{F}$ a Cd dimensional vector (i.e. a centralized supervector) by concatenating all $\tilde{F}_c$ and finally, let $\tilde{S}$ be the $Cd \times Cd$ diagonal matrix, whose C diagonal block are $\{\tilde{S}_c\}_{c=1}^C$.

To calculate the complete-data *likelihood* of a model with fixed parameters w given y_u, the following closed form expressions can be utilized

$$\log p(z_u|w) = G + H(w) \tag{13}$$

where

$$G = -\frac{1}{2}\text{tr}\left(\Sigma^{-1}\tilde{S}\right) - \sum_{c=1}^{C} N_c \log|2\pi\Sigma_c|^{1/2} \tag{14}$$

and

$$H(w) = w^T V^T \Sigma^{-1}\left(\tilde{F} - \frac{1}{2}\mathbf{N}Vw\right) \tag{15}$$

and $z_u = (y_u, \gamma_u)$ the (estimated) complete data.

The *posterior* distribution of w given z_u is also Gaussian $w \sim \mathcal{N}(\tilde{w}, \Lambda^{-1})$, where

$$\tilde{w} = \Lambda^{-1} V^T \Sigma^{-1} \tilde{F} \tag{16}$$

and

$$\Lambda = I_p + V^T \Sigma^{-1} \mathbf{N} V \tag{17}$$

the precision matrix of the posterior. Recall that $w \sim \mathcal{N}(0, I_p)$ a priori.

Finally, the *marginal* likelihood $p(z_u|S=1)$ is given by the following formula

$$\log p(z_u|S=1) = \log \int p(z_u|w, S=1)p(w)dw = G - \frac{1}{2}|\Lambda| + \frac{1}{2}\tilde{F}^T\Sigma^{-1}V\Lambda^{-1}V^T\Sigma^{-1}\tilde{F} \tag{18}$$

The existence of the above closed-form expressions is a consequence of using the (estimated) complete-data likelihood instead of the incomplete-data likelihood.

4.3 The VB algorithm

In order to solve the intractable problem of estimating $\mathcal{I}$ and S, a VB can been developed. Assume again that the variational posterior that can be factorized as $Q(Y, \mathcal{I}) = Q(Y)Q(\mathcal{I})$. Note though that in this setting, all the posteriors are conditional on (i) the complete-data $\{z_m\}_{m=1}^{M}$ and (ii) on a point-estimate of π. To update this estimate a further step should be added to the general VB-EM iteration, which is the maximization of the marginal likelihood w.r.t π. We initialize our variables by setting K equal to the maximum number of speaker K_{max}, and by setting π as uniform, i.e. $\pi_m = \frac{1}{K}, m = 1, \ldots, M$.

The E-step is responsible for estimating the assignment $\mathcal{I}$ given the current posterior distribution of the parameters $\{w_k\}_{k=1}^{K}$ and the current point-estimate π. Note that due to the conditioning on π, the factorization $Q(\mathcal{I}) = \prod_{m=1}^{M} Q(i_m)$ ($\{i_m\}_{m=1}^{M}$ are conditionally i.i.d.). Using the general update rule in (3) and after some matrix algebra, we end-up with

$$\text{VB-E step: } Q(i_m) = \prod_{k=1}^{K} q_{mk}^{i_{mk}}, \text{ where } q_{mk} = \frac{\tilde{q}_{mk}}{\sum_{k'=1}^{K} \tilde{q}_{mk'}} \tag{19}$$

and

$$\tilde{q}_{ms} = \pi_k p(z_m|\tilde{w}_k) \exp\left(-\frac{1}{2}\text{tr}\left(V^T \mathbf{N}_m \Sigma^{-1} V \Lambda_k^{-1}\right)\right) \tag{20}$$

where $p(z_m|\tilde{w}_k)$ and Λ_k are given in (13) and (17), respectively. Both quantities are estimated during the M-step of the previous iteration. Moreover, note as $\text{tr}\left(\Lambda_k^{-1}\right) \rightarrow 0$, i.e. no uncertainty is assumed regarding the estimates, the E-step degenerates to the corresponding step of the MAP-EM.

Similarly, the VB-M step is given according to the general rule in (4). After some matrix algebra we obtain

$$\text{VB-M step: } Q(w_k) \sim \mathcal{N}\left(\tilde{w}_k, \Lambda_k^{-1}\right) \tag{21}$$

i.e. will be a normal distribution with mean $\tilde{w}_k$ and precision Λ_k given in (16) and (17), respectively. Note again that the M-step of the MAP-EM is recovered by letting $\text{tr}\left(\Lambda_k^{-1}\right) \rightarrow 0$.

Finally, the additional step for re-estimating π is derived by maximizing the marginal likelihood w.r.t π. By rejecting irrelevant terms, the maximization problems becomes the following

$$\hat{\pi} = \arg\max_{\pi} \sum_{m=1}^{M} \sum_{k=1}^{K} q_{mk} \log \pi_k, \text{ subject to } \sum_{k=1}^{K} \pi_k = 1 \tag{22}$$

which yields

$$\pi \text{ update step: } \hat{\pi}_k = \frac{1}{M} \sum_{m=1}^{M} q_{mk} \tag{23}$$

By iteratively applying (19), (21) and (23) the algorithm converges to a maximum. After convergence is reached, the assignment of segments to speakers $\mathcal{I}$ and consequently the number of speakers K are estimated from (19) by simply assigning the mth segments to the speaker that maximizes $Q(i_m)$.

4.4 Experiments

So far, the proposed system has been tested only against telephone conversation datasets. This setting differs from the usual diarization systems, since we a priori know the number of speakers (i.e. $K = 2$). Therefore, the strength of the proposed VB-system as a model selection tool cannot be assessed from this series of experiments. However, the results show a drastic reduction in terms of Diarization Error Rate (DER,%). In Table 1, the DER on NIST 2008 SRE Summed Channel Test Data made by the VB-system are presented, for several front-end features. Details about the features can be found in (Kenny et al., 2010). The most successive front-end configuration includes 20 static-only MFCC, a 1024-component UBM and a gender-independent factor analysis model with 300 eigenvoices. The 2$^{\text{nd}}$ pass means that the speaker change points found by Viterbi resegmentation were used to initialize a second run of Variational Bayes and this was followed by another Viterbi resegmentation.

	configuration	*mean* DER(%)	σ(%)
	BUT features		
1	VB without Viterbi	9.1	11.9
2	VB with Viterbi	4.5	8.5
3	VB with Viterbi and 2$^{\text{nd}}$ pass	3.8	7.6
	CRIM features		
4	VB with 2$^{\text{nd}}$ pass, no Viterbi	3.3	7.8
	Raw cepstral features		
5	VB with 2$^{\text{nd}}$ pass, no Viterbi	2.2	5.8
6	VB with 2$^{\text{nd}}$ pass, no Viterbi	1.9	5.6

Table 1. DER (%) NIST 2008 SRE Summed Channel Test Data using the VB-system. The standard deviation of the Diarization Errors is denoted by σ.

In Table 1, the best performance of the VB-system is compared to (i) a baseline diarization system (i.e. speaker change detector, BIC-based AHC with single Gaussians and Viterbi resegmentation) augmented by a soft-clustering postprocessing stage, and (ii) a streaming system that operates on speaker factors and was introduced in (Castaldo et al., 2008) as a stream-based approach to performs online diarization. The conversation is seen as a stream of fixed-duration time slices and the system operates in a causal fashion. Speakers detected in the current slice are compared with previously detected speakers to determine if a new speaker has been detected or previous models should be updated. Further details about the implementation may be found in (Kenny et al., 2010).

System	*mean DER(%)*	σ(%)
Baseline with soft-clustering	3.5	8.0
Streaming with Viterbi	4.6	8.8
VB with raw cepstra, Viterbi and 2$^{\text{nd}}$ pass	1.0	3.5

Table 2. Best results obtained on the NIST 2008 SRE Summed Channel Telephone Data using the baseline, the streaming and the VB systems.

5. An HMM-based approach using hierarchical dirichlet processes

In this chapter, we present a recent SD approach that is based on Bayesian nonparametric modeling, (Fox et al., 2009). This approach utilizes the HMM framework to model the inter-speaker dynamics, mixture models for the emission probabilities and (averages of) MFCC as front-end-features. Its main contribution is the use of infinite models on both of the HMM levels, i.e. on the multimodal emission probabilities and on the states and the transitions between them.

5.1 General about infinite models

The use of infinite models is a natural way to overcome the issue of how to determine a priori the order of a model. First, consider the problem of determining the order of the GMM that should be used to model the distribution of a speaker. A fully Bayesian modeling should consider the order of the model as a random variable, and treat it in the same way it treats the rest of the parameters; the order should be integrated out, too, just like the weights, the means and the covariance matrices. On the HMM-level, a classical approach to determine the number of states K is to apply Viterbi, Baum-Welch or VB-EM type of learning for each of the candidate K by conditioning on K (i.e. on the hypothesis), and select the order that maximizes the evidence (or an approximation) of the model. The Evolving-HMM and the VB approach of (Valente, 2005) are typical examples of this framework. However, such exhaustive search solutions may lack of efficiency, especially in cases where the hypothesis space is quite large (e.g. Broadcast News). A more flexible solution is offered by infinite HMM, where the number of states are not specified a priori, but is rather inferred in a more data-driven way.

5.2 Infinite mixture models and the Dirichlet processes

We begin the analysis by describing the Dirichlet process (DP), which is the building block in most of the infinite models, (Ferguson, 1973). The DP can be considered as a infinite extension of the Dirichlet distribution. In the same way the Gaussian process can be utilized in Bayesian inference as a prior (i.e a measure) on functions, the DP can be used as a measure on measures. Moreover, much like the derivation of the familiar Gaussian process from the Gaussian distribution, the DP may be explicitly derived from the Dirichlet distribution by letting its order go to infinity.

Let us assume that $\beta = \{\beta_k\}_{k=1}^{K}$ follows a symmetric Dirichlet distribution of order K and strength α_0, i.e.

$$\beta|\alpha_0 \sim \mathcal{D}ir(\underbrace{\alpha_0/K, \ldots, \alpha_0/K}_{K \text{ components}}) \tag{24}$$

where $0 < \beta_k < 1, k = 1, \ldots, K$ and $\sum_{k=1}^{K} \beta_k = 1$. Suppose we aim to construct a generative model for GMMs. Then, β can be used as the weights of the model. We also need an appropriate *base* measure G_0. In the DP-GMM case, G_0 is the prior distribution of the components, e.g. a Normal-Inverse Wishart distribution if conjugacy is desired. By sampling the measure G_0 K times, i.e. $\theta_k \sim G_0, k = 1, \ldots, K$ we get a set of K *atoms* that can be associated with $\{\beta_k\}_{k=1}^{K}$. The distribution of θ given $\{\beta_k, \theta_k\}_{k=1}^{K}$ can expressed as $\theta|G^k \sim G^k$ where

$$G^K = \sum_{k=1}^{K} \beta_k \delta_{\theta_k} \tag{25}$$

where $\delta_{\theta_k} = \delta(\cdot, \theta_k)$. The distribution G^K can now be used in order to generate random samples from G^K. One should first sample $\theta|G^K \sim G^K$ and then sample $y|\theta \sim F(\theta)$, where $F(\cdot)$ is the Gaussian distribution.

Suppose now that we let $K \to \infty$. Then, $\beta|\alpha_0$ follows a DP with concentration parameter α_0. The random draw from the DP becomes an infinite mixture, i.e.

$$G = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k} \tag{26}$$

We say that G follows a DP and we denote it by

$$G \sim \mathrm{DP}(\alpha_0, G_0) \tag{27}$$

Let us consider N samples from G, denoted by $\{\phi_n\}_{n=1}^N$, $\phi_n|G \sim G$. What prevents K from going to infinity as $N \to \infty$ is a fundamental property of the Dirichlet distribution $\mathcal{D}ir(\{g_k\}_{k=1}^K)$. Starting from $g_k = 1, k = 1, \ldots, K$ and letting $g_k \to 0$, the probability mass is being increasingly concentrated on areas close to the K *vetrices* of the $(K-1)$-simplex. Hence, even if $N \to \infty$, G remains discrete and the cardinality of the set finite.

The posterior of G, i.e. conditioned to a set $\{\phi_n\}_{n=1}^N$ is a DP, parametrized as follows

$$G \sim \mathrm{DP}\left(\alpha_0 + N, \frac{1}{\alpha_0 + N}\left[\alpha_0 G_0 + \sum_{n=1}^{N} \delta_{\phi_n}\right]\right) \tag{28}$$

or equivalently

$$G \sim \mathrm{DP}\left(\alpha_0 + N, \frac{1}{\alpha_0 + N}\left[\alpha_0 G_0 + \sum_{k=1}^{K} N_k \delta_{\theta_k}\right]\right) \tag{29}$$

where $N_k = \sum_{n=1}^{N} \delta(\phi_n, \theta_k)$ and $\sum_{k=1}^{K} N_k = N$.

It order to create samples from the DP, we may proceed as follows

$$\phi_{N+1}|\{\phi_n\}_{n=1}^N \sim \frac{1}{\alpha_0 + N}\left(\alpha_0 G_0 + \sum_{n=1}^{N} \delta_{\phi_n}\right) \tag{30}$$

i.e. there is no need to refer to G. What (30) shows is that as N grows, the probability of getting previously unseen samples decreases linearly. Furthermore, the probability of the new sample to be equal to θ_k is equal to $(\alpha_0 + N)^{-1} N_k$. Finally, high values of α_0 corresponds to high rates of generating unseen atoms.

Given α_0, the prior of the number of distinct atoms K after N samples is given by

$$p(K|\alpha_0, N) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_0 + N)} s(N, K) \alpha_0^K \tag{31}$$

where $s(N, K)$ are unsigned Stirling numbers of the first kind.

A intuitive and constructive interpretation of β is the *stick-breaking* process, (Sethuraman, 1994). For the finite case, we saw that β follows the Dirichlet distribution. In order to create samples of β for the infinite case, however, the following sampling scheme is useful. Considering a stick of unitary length. For $k = 1, 2, \ldots,$

$$u_k|\alpha_0 \sim \mathcal{B}eta(1, \alpha_0) \tag{32}$$

$$\beta_k = u_k \left(1 - \sum_{k'=1}^{k-1} \beta_{k'}\right) = u_k \prod_{k'=1}^{k-1} (1 - u_{k'}) \tag{33}$$

Therefore, u_k is distributed as $\mathcal{B}eta(1, \alpha_0)$ and covers a fraction of u_k of the remaining stick. Hence, the overall length that covers is equal to β_k, given by (33). Note that a usual notation from the stick-breaking weights is $\beta \sim \text{GEM}(\alpha_0)$, where GEM stands for Griffiths, Engen and McCloskey. The generative model is depicted in Fig. 2(a).

5.3 Infinite Hidden Markov Models and the hierarchical DP

Let us now examine how can we apply similar ideas to a dynamic network, namely the (time-independent) HMM. An HMM can be considered as a collection of GMMs, that differ only on their *weights* which correspond to the *rows* of the transition matrix A. Each row $A_{k\cdot} = [A_{k1}, \ldots, A_{kK}]$, $k = 1, \ldots, K$ is the conditional probability of $x^{t+1} = l$, $l = 1, \ldots, K$ given $x^t = k$. Moreover, the initial probabilities $a = [a_1, \ldots, a_K]$ may also be treated in a similar way, by defining the non-emitting zero state. This allows us to include all the transition parameters in a unique matrix, defined as the augmented transition matrix $A^+ = [a^T, A^T]^T$.

In the finite-state case, a standard Bayesian strategy is to place a common prior on each line of A, e.g.

$$p(A_{k\cdot}|\gamma) = \mathcal{D}ir(\gamma/K, \ldots, \gamma/K) \tag{34}$$

Two are the drawbacks of this approach. The first is that the state-persistence that several dynamic systems exhibit is not captured explicitly in this prior. As we show next, this can be solve rather easily, by adding an extra hyperparameter to the diagonal of A that is capable of biasing the dynamics towards self-transition. A further and more severe in our case drawback is that such a prior cannot be extended to the infinite case. This is because the tying between the weights $A_{k\cdot} = [A_{k1}, \ldots, A_{kK}]$, $k = 1, \ldots, K$ that is offered by placing a common prior is weak when $K \to \infty$. What this prior implies is that each $A_{k\cdot}$ should simply be a independent draw from a DP having a common concentration parameter γ and a common continuous base measure H. Hence, the set of atoms between every pair of draws would be disjoint, leading to no sparse solutions at all. As proposed in (Teh et al., 2006), what is required to tackle this

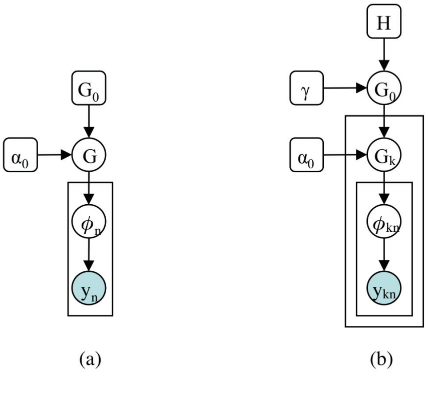


Fig. 2. Plate notations of the DP-mixtures. (a) The original DP-mixture model, (b) The Hierarchical DP-mixture model

problem is to add another level in the hierarchy. On the uppermost level, a single draw G_0 from $DP(\gamma, H)$ is generated, i.e.

$$G_0|\gamma, H \sim DP(\gamma, H) \tag{35}$$

This draw is then used to parametrize the DP prior of each of the states, i.e.

$$G_k|\alpha_0, G_0 \sim DP(\alpha_0, G_0),\ k = 1, \dots, \infty \tag{36}$$

The generative model is depicted in Fig. 2(b). Contrary to the previous approach, the base measure of G_k (denote by G_0) is not only common to all states, but is moreover discrete, since

$$G_0|\gamma, H = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k} \tag{37}$$

Hence, each G_k would be a (weighted) collection of the same set of atoms. Moreover, not only the set is the same, but identically weighted by $\beta = \{\beta_k\}_{k=1}^{\infty}$. Using the stick-breaking construction, each row of the transition matrix is distributed as follows. For $k = 1, 2, \dots$ and $k' = 1, 2, \dots$

$$u_{kk'}|\alpha_0 \sim \mathcal{B}eta\left(\alpha_0\beta_{k'}, \alpha_0\left(1 - \sum_{l=1}^{k'} \beta_l\right)\right) \tag{38}$$

$$A_{kk'} = u_{kk'} \prod_{l=1}^{k'-1} (1 - u_{kl}) \tag{39}$$

The expected values of each $A_{k\cdot}$ will be equal to β_k. Moreover, the concentration α_0 now controls both the state-connectivity and the similarity between each $A_{k\cdot}$. High values of α_0 means that most of the samples will be generated directly from G_0, which increases the state connectivity and decreases the variability between $\{A_{k\cdot}\}_{k=1}^{K}$. Contrarily, for low values of α_0 the HMM may exhibit sparse state-connectivity, i.e. each state may be accessible only via a subset of the other states.

5.4 Hierarchical DP HMM with DP mixture models as emission probabilities

Let us recapitulate the above modeling. We showed that the Hierarchical DP is a natural extension of the original DP, that is suitable in cases where the overall model is decomposed to a collection of submodels that share some certain properties. HMMs are such models, since they can be considered as collections of conditional mixtures, where the conditioning is w.r.t. the current state. We emphasize that these mixtures should not be confused with the possibility of modeling the emissions probabilities with mixture models. The emission probabilities are governed completely by the base measure $H(\cdot)$. If we desire to include finite mixtures (e.g. GMMs) then $H(\cdot)$ should be the Dirichlet-Normal-Inverse Wishart prior distribution, if conjugacy is desired. The tth observation y^t will then follow a GMM distribution, $y^t|\theta_{x^t} \sim F(\theta_{x^t})$. Thus, each atom θ_k will be a parametrization of a GMM, capable of describing the multimodal distribution of a speaker.

For describing the distribution of a speaker, the use of DP-mixture models may be considered as well. This means that both the HMM and its emissions may be considered as infinite (i.e. nonparametric), which is the method proposed in (Fox et al., 2009). However, in order to avoid fast transitions between states, a bias towards self-transitions is adopted, that allows to distinguish between the underlying HDP-HMM states and the within-speaker multimodal emissions. Moreover, non-overlapping 250*ms* frames are used as front end features while a minimum duration of 500*ms* is imposed on speaker segments. The resulting model, termed as the *Sticky*-HDP-HMM produced state-of-the-art results even without any prior tuning. In fully Bayesian approaches, tuning is related to the hyperparameters of the uppermost layer. We also emphasize that the use of infinite models is SD has previously been proposed in (Valente, 2006). It uses a DP-mixture model for the emissions and an Infinite-HMM for modeling the transition dynamics. However, the HMM used was degenerated (all rows of A are assumed to be equal) making the hierarchical DP unnecessary. A VB algorithm was proposed, based on the mean-field approximation, while a slight improvement was reported over the baseline VB with finite mixtures.

5.5 Inference

Methods of inference of the Sticky-HDP-HMM is out of the scope of this review. The interested reader is encouraged to examine the inferential procedures given in (Teh et al., 2006) and (Fox et al., 2009).

In general, to infer such models, the most usual way is the family of Markov Chain Monte Carlo (MCMC) methods. Like any sampling method, MCMC aims to estimate any desired quantity by sample averages, generated according a proper measure. In cases where all of the distributions are conjugate to their priors, Gibbs sampling is usually a sufficient and easy to implement MCMC method. It proceeds with sampling each random variable, conditioned on all the others, which are set to their current values. The Gibbs sampler is not a unique technique in the models described above. This is because there are alternative generative models by which the same process can be defined. Several Gibbs samplers have been proposed, that vary according to their mixing rates and their implementation effort that is required. A detailed implementation of these such samplers, along with a comparison between them can be found in (Teh et al., 2006). Other approaches, that are better suited the HMM framework are presented in (Fox et al., 2009). Finally, we mention the possibility of applying variational inference to infinite models. Such approaches are analyzed in (Blei & Jordan, 2005) and (Valente, 2006) and can be much faster that MCMC.

5.6 Experiments

The experiments of the Sticky-HDP-HMM presented in (Fox et al., 2009) are based on the NIST-2007 meeting data and are being compared to (i) the non-sticky-HDP-HMM and to (ii) the ICSI diarization system, (Wooters & Huijbregts, 2008). The latter system is based on AHC and was the winner of the competition, scoring a 18.37% DER. It uses ML-GMMs to model the emission probabilities, a penalty free BIC-like approach and a Viterbi algorithm after each cluster merging. The comparison between the two HDP systems is presented in Table 3. The number in the parentheses is the performance when running the 16^{th} meeting for 50,000 Gibbs iterations, instead of the fixed number of 10,000 iterations. The results clearly show the usefulness of the state persistence parameter in avoiding the unrealistic fast transitions between speakers that is translated to an approximate 20% relative improvement in DER.

Overall DERs (%)	Min Hamming	Max Likelihood	2-Best	5-Best
Non-Sticky HDP-HMM	23.91	25.91	23.67	21.06
Sticky HDP-HMM	19.01 (17.84)	19.37	16.97	14.61

Table 3. Best results obtained on the NIST-2007 Meeting Data using the Sticky and the Non-Sticky HDP-HMM.

Compared to the ICSI system, the Sticky-HDP-HMM performed slightly worst, if we consider the setting with 10.000 iterations. We should note though that no tuning has been applied, i.e. the priors on the hyperparameters are very vague, and are therefore placing significant prior mass over areas that are unrealistic for the specific application field. Hence, by assuming a proper tuning of the uppermost hyperperameters, a further increase in the accuracy should normally be expected.

Note finally, that due to the fully Bayesian paradigm, several alternative state-sequences may be sampled from the posterior. As Table 3 shows, if the best per-meeting DER for the five most likely samples is considered, our overall DER drops to 14.61%. Finally, the possibility of providing multiple state-sequences, along with their posterior probability mass, is a desirable property when applying fusion techniques. In such cases, the relative uncertainty of the decisions made by each information stream should also be assessed in order to fuse the streams in a fully probabilistic manner.

6. Conclusions and further research directions

In this chapter, we presented a introduction to some of the recent methods that have been proposed in SD. We restricted ourselves to some novel fully-Bayesian approaches, that are based on (i) finite mixtures with Variational Bayes inference methods, and (ii) nonparametric (i.e. infinite) Bayesian approaches. These methods are applicable to numerous problems that deal with clustered data and are gaining increasing attention in several fields. We analyzed some of the theoretical advantages over non- or semi-Bayesian approaches and their strength and flexibility in learning the clustered structures of the data.

Bayesian nonparametrics may be used to tackle several other tasks in speaker and audio problems, as well. For example, speaker verification is another major task that can be treated as a model selection problem (that is one versus two speakers), and the effectiveness of fully-Bayesian approaches has recently been proven, (Kenny, 2010). Furthermore, SVM-based verification is a field where Bayesian approaches can be examined. A severe problem with SVMs is that their soft-outputs cannot be regarded as probabilistic. On the contrary, relevance vector machines (RVM) are fully-probabilistic analogues to SVMs and as such they can be used as an alternative discriminative framework, (Tipping, 2001). Speaker separation from multiple (or single) microphones is another related task to SD. A Bayesian nonparametric model, termed as infinite factorial HMM has been used to separate the speakers and infer their number, (Van Gael et al., 2009). Such approaches can be used in SD as well, in order to detect and identify overlapping speakers. Finally, several inference methods can be tested in speaker technologies, such as the Annealing Importance Sampling (Neal, 1998) and Expectation-Propagation (Minka, 2001) that produce state-of-the-art results in many other fields.

7. Appendix: Super- and i-vectors feature spaces

We review here some of the new feature spaces that are used in most of the contemporary speaker verification systems and recently in several SD systems as well. These features are derived by (i) adapting a UBM with the observation vectors of a speech segment (using the standard EM-MAP of (Reynolds et al., 2000)) and (ii) mapping the high-dimensional concatenated mean vector (or *supervector*) to a mid-dimensional subspace, resulting in the identity vector, or simply the *i-vector*. The transformation rule is derived offline, using enrollment data, and aims to reduce the dimensionality of the new feature space, while

discarding those directions that do not carry speaker discriminant information.
The major advantage of the new feature space is the mapping of variable-length utterances onto a space of fixed dimensionality, through a well-tested statistical intermediate description (i.e. the UBM-based adaptation scheme). Using the i-vector representation, several kernel-based and other general purpose algorithms can be applied in order to perform identification, verification, and clustering. Finally, the i-vectors of a speaker have a rather Gaussian distribution since they represent mean values, projected onto a lower dimensional basis and they take values on $\Re^p$. Hence, several algorithms that have been developed assuming Euclidean spaces (i.e. of constant metric tensor) can be applied without much adaptation. This is in contrast to representations that lie on spaces where the natural statistical divergences (e.g. KL, Hellinger) have complex expressions that are far from being (squared) Euclidean distances, such as those that include weights or covariance matrices.

MAP-estimate based on UBM and the supervector representation

As discussed in section 2, a typical preprocessor applies MFCC extraction, delta-feature calculation and voice activity detection. When performing speaker verification, normalization methods such as mean and variance normalization, RASTA filtering and feature warping are essential in order to compensate for the channel-effects, (Kinnunen & B, 2010).
An effective statistical representation of the stream $Y = \{y^t\}_{t=1}^{T}$ of front-end features is a Gaussian Mixture Model (GMM). The model, however, is not trained from scratch. Instead of a Maximum Likelihood (ML) estimate, the observations are used to adapt a well-trained model (Universal Background Model, UBM) with parameters $\lambda_{ubm} = \{\pi_c^0, \mu_c^0, \Sigma_c^0\}_{c=1}^{C}$ that denote weights, means and (diagonal) covariance matrices, respectively. The UBM is a GMM that is trained offline with the standard ML-EM algorithm, using hours of speech data and a huge number of speakers. The final estimate $\tilde{\lambda}_Y$ of the p.d.f. of Y is the Maximum A Posteriori (MAP) estimate of λ_Y, and is calculated by a MAP-Expectation-Maximization (MAP-EM) algorithm. Moreover, only the mean-values are allowed to be adapted, which implies that the mean values $\{\tilde{m}_c\}_{c=1}^{C}$ are sufficient to represent the model $\tilde{\lambda}_Y$ for a fixed UBM.
The E-step of the ith iteration is carried as

$$P(c|y^t, \tilde{\lambda}_Y^{(i-1)}) = \frac{\pi_c^0 p(y^t|\tilde{\mu}_c^{(i-1)}, \Sigma_c^0)}{\sum_{c=1}^{C} \pi_c^0 p(y^t|\tilde{\mu}_c^{(i-1)}, \Sigma_c^0)} \tag{40}$$

followed by the corresponding M-step

$$\tilde{\mu}_c^{(i)} = \alpha_c^{(i)} \bar{y}_c + (1 - \alpha_c^{(i)}) \mu_c^0 \tag{41}$$

where

$$\alpha_c^{(i)} = \frac{n_c^{(i)}}{n_c^{(i)} + r} \tag{42}$$

$$n_c^{(i)} = \sum_{t=1}^{T} P(c|y^t, \tilde{\lambda}_Y^{(i-1)}) \tag{43}$$

and

$$\bar{y}_c = \frac{1}{n_c^{(i)}} \sum_{t=1}^{T} P(c|y^t, \tilde{\lambda}_Y^{(i-1)}) y^t \tag{44}$$

The above expressions reveal that λ_{ubm} and r are completely specifying the *prior* of $\{\mu_c\}_{c=1}^{C}$. Its density is as follows

$$\mu_c | r, \Sigma_c \sim \mathcal{N}\left(\mu_c^0, \frac{1}{r}\Sigma_c^0\right) \tag{45}$$

where $\mathcal{N}(\mu, \Sigma)$ denotes the normal p.d.f., with mean and covariance matrix μ and Σ respectively. The parameter r corresponds to the *strength* of the prior of $\{\mu_c\}_{c=1}^{C}$, i.e. the equivalent number of virtual observations that are backing the initial estimate μ_c^0.

Apart from the increase in the robustness of the estimate of λ_Y, a further severe benefit from using a UBM is the common *ordering* that it establishes to the C open areas of the observations' space. Consider the MAP estimates $\tilde{\lambda}_{Y_a}$ and $\tilde{\lambda}_{Y_b}$ given Y_a and Y_b respectively. Due to their common initialization by λ_{ubm}, $\tilde{\lambda}_{Y_a}$ and $\tilde{\lambda}_{Y_b}$ are directly comparable, in the sense that their corresponding entries carry information about the same a priori area of the observations' space, apart from the dimension. Such a correspondence cannot be achieved when the models are trained using ML-EM algorithm, making several fast scoring methods and dimensionality reduction techniques inapplicable. The concatenated vector $M_Y \in \Re^{Cd}$ of the means $\{\tilde{\mu}_c\}_{c=1}^{C}$ is termed *supervector* and can be considered as a novel fixed-size way for representing Y.

Likelihood ratios in verification and clustering

A standard way to score a new set of observation against a model $\tilde{\lambda}_s$ is based on the normalized log-likelihood ratio between the $\tilde{\lambda}_s$ and λ_{ubm}, i.e.

$$NLLR(\tilde{\lambda}_s | Y, \lambda_{ubm}) = \frac{1}{T}\sum_{t=1}^{T} \log \frac{p(y^t|\tilde{\lambda}_s)}{p(y^t|\lambda_{ubm})} \tag{46}$$

The coupling between $\tilde{\lambda}_s$ and the UBM increases drastically the robustness of the ratio, and allows fast scoring methods to be applied.

The NLLR can be deployed in order to apply both verification and clustering. In verification, NLLR is normalized properly according to a set of cohort speakers and then a simple threshold is applied to verify the claimed identity. Several score-level normalization methods have been proposed (e.g. z-norm, t-norm, s-norm) and are aiming to compensate the speaker and channel dependent behavior of the statistic NLLR.

In most step-by-step SD approaches, UBM-based models are used only after a first clustering pass with single-Gaussian models. The clusters that are created are then used to initialize further iterations of UBM-based hierarchical clustering. To define a similarity measure between two clusters Y_a and Y_b, the Cross Likelihood Ratio (NCLR)

$$CLR(Y_a, Y_b) = NLLR(\tilde{\lambda}_a | Y_b, \lambda_{ubm}) + NLLR(\tilde{\lambda}_b | Y_a, \lambda_{ubm}) \tag{47}$$

and the Normalized Cross Likelihood Ratio (NCLR)

$$NCLR(Y_a, Y_b) = \frac{1}{T_a}\sum_{y^t \in Y_a} \log \frac{p(y^t|\tilde{\lambda}_b)}{p(y^t|\tilde{\lambda}_a)} + \frac{1}{T_b}\sum_{y^t \in Y_b} \log \frac{p(y^t|\tilde{\lambda}_a)}{p(y^t|\tilde{\lambda}_b)} \tag{48}$$

are both symmetric measures that have been applied successfully, (*see* (Le et al., 2007) for a comparison). However, a predefined threshold is required to decide whether a pair of clusters should be merged or not, (Zhu et al., 2005).

Kernels based on supervectors

One of the drawbacks of a likelihood ratio-based verification and clustering algorithms is their dependence on the data Y. This problem arises from the fact that the likelihood function of the *incomplete* data

$$p(y|\tilde{\lambda}) = \sum_{c=1}^{C} \pi_c p(y|\mu_c, \Sigma_c) \tag{49}$$

does not belong to an *exponential* family and, therefore, a sufficient statistic does not exist, (Wainwright & Jordan, 2008). On the contrary, the *complete-data likelihood*, i.e. the likelihood of $z^t = (x^t, y^t)$ where $X = \{x^t\}_{t=1}^T$ denotes the alignment of Y to components - belongs to the exponential family

$$p(x, y|\tilde{\lambda}) = \sum_{c=1}^{C} \delta(c, x)\pi_c p(y|\mu_c, \Sigma_c) \tag{50}$$

and therefore several closed-form expressions can be utilized. The obvious problem is that we do not know x^t. However, their MAP estimate $\tilde{x}^t$ of x^t is already in-hand, from the last E-step of the EM algorithm. This is the rationale for the use of similarity measures between utterances that are not based on likelihood-ratios. In (Campbell, Sturim & Reynolds, 2006), a KL divergence-like kernel that is proposed

$$K(\tilde{\lambda}_a, \tilde{\lambda}_b) = \sum_{c=1}^{C} \left(\sqrt{\pi_c}\Sigma_c^{-1/2}(\tilde{\mu}_c^a - \mu_c^0)\right)^T \left(\sqrt{\pi_c}\Sigma_c^{-1/2}(\tilde{\mu}_c^b - \mu_c^0)\right) \tag{51}$$

Such kernels implicitly make use of the complete-data likelihood, and the corresponding closed-form expressions. Once the kernel is defined, one may consider the use of Support Vector Machines (SVMs) to perform verification. During training, the separating hyperplane should be estimated, based on a labeled training set that consists of both positive and negative examples $\{\tilde{\lambda}_i, t_i\}_{i=1}^N$, where $t_i \in \{-1, +1\}$. During verification, a sparse subset Λ_s of these examples (i.e. the support vectors) $\{\tilde{\lambda}_i, t_i\}_{i\in\Lambda_s}$ along with their weights $\{\alpha_i\}_{i\in\Lambda_s}$ and the bias term b are needed to perform verification, according to $sgn(f(\tilde{\lambda}'))$, where

$$f(\tilde{\lambda}') = \sum_{i\in\Lambda_s} \alpha_i t_i K(\tilde{\lambda}', \tilde{\lambda}_i) + b \tag{52}$$

denotes the function that defines the hyperplane. Several other kernels and additional information regarding the SVM-based verification can be found in (Campbell, Campbell, Reynolds, Singer & Torres-Carrasquillo, 2006).

From supervectors to i-vectors

In practice, the dimensionality of supervectors is very large to handle (e.g. $dim(M_Y) = 77824$ for $(d, C) = (38, 2048)$). Therefore, it is a natural field for applying dimensionality reduction (DR) methods. A common method for DR is Principal Component Analysis (PCA). The eigenvectors having the highest corresponding eigenvalues are termed *eigenvoices*, inspired from the similar concept of eigenfaces in face recognition, (Turk & Pentland, 1991).

However, PCA is an *unsupervised* method, and as such, it does not take into account neither the clustered structure of the enrollment data nor the classification purpose of the DR. Linear Discriminant Analysis (LDA) is a popular *supervised* method for defining such bases and is the one that is used to extract the i-vectors. The supervector M (of $\kappa = Cd$ dimensions) is assumed to be generated from the following equation

$$M = M_0 + Tw + e \tag{53}$$

where M_0 the supervector of the UBM, T a $(\kappa \times p)$-dimensional matrix (where $p \ll \kappa$, typically $p = 400$), w a p-dimensional vector having a standard normal distribution, i.e. $w \sim \mathcal{N}(0, I_p)$ and e the approximation error. The matrix T is called *total variability matrix* and its columns are forming the LDA-derived subspace with which M is expressed. The term total variability matrix stems from the fact that the labeling used in LDA treats each speaker recording (i.e. each set of utterances of a speaker from the same session) as a distinct class, (Dehak et al., 2011). This strategy is in contrast to a former one, that applies Joint-factor Analysis (JFA) to model separately between-speaker and within-speaker variability.

To calculate the i-vector w of an utterance u that consists of Y, assuming a UBM and a basis T, one should (i) adapt the UBM using the standard MAP-adaptation scheme, and (ii) use the centralized mean vectors to calculate the i-vector with the following formula

$$w = \left(I_p + T^T \Sigma_e^{-1} N_u T \right)^{-1} T^T \Sigma_e^{-1} F_u \tag{54}$$

In (54), F_u denotes the centralized supervector of the utterance, i.e. $F_u = M_u - M_0$, N_u a $\kappa \times \kappa$ diagonal matrix, whose K diagonal blocks are defined as $n_c I_d$ and n_c given in (43), and finally Σ_e a $\kappa \times \kappa$ diagonal covariance matrix, estimated during LDA, that models the expected variance of the approximation error e.

These vectors may be considered as lying on a feature space that is well suited to tasks like speaker verification, identification and diarization.

8. References

Ajmera, J. & Wooters, C. (2003). A robust speaker clustering algorithm, *Proc. ASRU*, St. Thomas, U.S. Virgin Islands, pp. 411–416.

Amari, S. (1995). Information geometry of the EM and em algorithms for neural networks, *Neural Networks* 8: 1379–1408.

Anguera, X., Wooters, C. & Hernando, J. (2007). Acoustic beamforming for speaker diarization of meetings, *IEEE TASLP* 15(7): 2011–2021.

Barras, C., Zhu, X., Meignier, S. & Gauvain, J. (2004). Improving Speaker Diarization, *Proceedings of Fall 2004 Rich Transcription Workshop (RT-04)*.

Blei, D. M. & Jordan, M. I. (2005). Variational inference for Dirichlet process mixtures, *Bayesian Analysis* 1: 121–144.

Campbell, W., Campbell, J., Reynolds, D., Singer, E. & Torres-Carrasquillo, P. (2006). Support vector machines for speaker and language recognition, *Computer Speech and Language* 20(2-3): 210 – 229. Odyssey 2004: The speaker and Language Recognition Workshop - Odyssey-04.

Campbell, W. M., Sturim, D. E. & Reynolds, D. A. (2006). Support vector machines using GMM supervectors for speaker verification, *IEEE Signal Processing Letters* 13: 308–311.

Castaldo, F., Colibro, D., Dalmasso, E., Laface, P. & Vair, C. (2008). Stream-based speaker segmentation using speaker factors and eigenvoices, *Proc. IEEE ICASSP*, pp. 4133–4136.

Chen, S. & Gopalakrishnam, P. (1998). Speaker, Environment and Channel Change Detection and Clustering via the Bayesian Information Criterion, *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*.

Corduneanu, A. & Bishop, C. M. (2001). Variational bayesian model selection for mixture distributions, *Eighth Int'l Conf. Artificial Intelligence and Statistics*, pp. 27–34.

Dehak, N., Kenny, P., Dehak, R., Dumouchel, P. & Ouellet, P. (2011). Front-end factor analysis for speaker verification, *Audio, Speech, and Language Processing, IEEE Transactions on* 19(4): 788 –798.

Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society Ser. B* 39.

Ferguson, T. S. (1973). A bayesian analysis of some nonparametric problems, *Annals of Statistics* 1: 209–230.

Fox, E. B., Sudderth, E. B., Jordan, M. I. & Willsky, A. S. (2009). The Sticky HDP-HMM: Bayesian nonparametric Hidden Markov Models with Persistent States.

Fraley, C. & Raftery, A. E. (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis, *Comput. J.* 41: 578–588.

Friedland, G., Vinyals, O., Huang, Y. & Müller, C. (2009). Prosodic and other long-term features for speaker diarization, *IEEE Transactions on Audio, Speech & Language Processing* 17(5): 985–993.

Gupta, V., Kenny, P., Ouellet, P., Boulianne, G. & Dumouchel, P. (2007). Combining Gaussianized/non-Gaussianized Features to Improve Speaker Diarization of Telephone Conversations, *IEEE Signal Processing Letters* 14(12): 1040–1043.

Hermansky, H., Hanson, B. A. & Wakita, H. (1985). Perceptually based linear predictive analysis of speech, *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 509–512.

Hermansky, H., Morgan, N., Bayya, A. & Kohn, P. (1992). RASTA-PLP speech analysis technique, *Acoustics, Speech, and Signal Processing, IEEE International Conference on* 1: 121–124.

Ishizuka, K., Nakatani, T., Fujimoto, M. & Miyazaki, N. (2010). Noise robust voice activity detection based on periodic to aperiodic component ratio, *Speech Commun.* 52: 41–60.

Kenny, P. (2010). Bayesian speaker verification with heavy tailed priors, *Computer Speech and Language* . Odyssey 2010: The speaker and Language Recognition Workshop - Odyssey-10, Brno, Czech Republic.

Kenny, P., Boulianne, G. & Dumouchel, P. (2005). Eigenvoice modeling with sparse training data., *IEEE Transactions on Speech and Audio Processing* 13(3): 345–354.

Kenny, P., Reynolds, D. & Castaldo, F. (2010). Diarization of telephone conversations using factor analysis, *Selected Topics in Signal Processing, IEEE Journal of* 4(6): 1059 –1070.

Kinnunen, T. & B, H. L. (2010). An overview of text-independent speaker recognition: from features to supervectors", speech communication.

Lapidot, I. (2003). SOM as likelihood estimator for speaker clustering, *in Proc. Eurospeech*.

Le, V.-B., Mella, O. & Fohr, D. (2007). Speaker Diarization using Normalized Cross Likelihood Ratio, *proceeding of INTERSPEECH 2007 INTERSPEECH 2007*, ISCA, Antwerp Belgium.

Mackay, D. J. C. (2003). Information theory, inference, and learning algorithms, *Cambridge University Press New York* .

Meignier, S., Moraru, D., Fredouille, C., Bonastre, J.-F. & Besacier, L. (2006). Step-by-step and integrated approaches in broadcast news speaker diarization, *Computer Speech and Language* 20: 303–330.

Minka, T. (2001). Expectation propagation for approximate bayesian inference, *Proceedings of the Proceedings of the Seventeenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-01)*, Morgan Kaufmann, San Francisco, CA, pp. 362–36.

Neal, R. M. (1998). Annealed importance sampling, *STATISTICS AND COMPUTING* 11: 125–139.

Ning, H., Liu, M., Tang, H. & Huang, T. (2006). A Spectral Clustering Approach to Speaker Diarization, *Proceedings of the International Conference on Spoken Language Processing*.

Pardo, J., Anguera, X. & Wooters, C. (2007). Speaker diarization for multiple-distant-microphone meetings using several sources of information, *IEEE Trans. Comput.* 56: 1189–1224.

Pelecanos, J. & Sridharan, S. (2001). Feature warping for robust speaker verification, *ODYSSEY-2001*, pp. 213–218.

Reynolds, D. A., Quatieri, T. F. & Dunn, R. B. (2000). Speaker verification using adapted Gaussian mixture models, *Digital Signal Processing*, Vol. 10, pp. 19–41.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors, *Statistica Sinica* 4: 639–650.

Snoussi, H. (2005). The geometry of prior selection, *Neurocomputing* 67: 214–244.

Stafylakis, T., Katsouros, V. & Carayannis, G. (2010a). The Segmental Bayesian Information Criterion and its applications to Speaker Diarization, *IEEE Selected topics in Signal Processing* pp. 857 – 866.

Stafylakis, T., Katsouros, V. & Carayannis, G. (2010b). Speaker clustering via the mean shift algorithm, *Odyssey 2010: The speaker and Language Recognition Workshop - Odyssey-10, Brno, Czech Republic*.

Teh, Y. W., Jordan, M. I., Beal, M. J. & Blei, D. M. (2006). Hierarchical Dirichlet processes, *Journal of the American Statistical Association* 101(476): 1566–1581.

Tipping, M. E. (2001). Sparse Bayesian Learning and the Relevance Vector Machine, *Journal of Machine Learning Research* 1: 211–244.

Tranter, S. & Reynolds, D. (2006). An overview of automatic speaker diarization systems, *IEEE Trans. Audio, Speech, and Language Processing* 14: 1557–1565.

Turk, M. A. & Pentland, A. P. (1991). Face recognition using eigenfaces, *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Comput. Sco. Press, pp. 586–591.

Valente, F. (2005). *Variational Bayesian methods for audio indexing*, PhD thesis.

Valente, F. (2006). Infinite models for speaker clustering, *International Conference on Spoken Language Processing*.

Van Gael, J., Teh, Y. W. & Ghahramani, Z. (2009). The infinite factorial hidden Markov model, *Advances in Neural Information Processing Systems*, Vol. 21.

Wainwright, M. J. & Jordan, M. I. (2008). *Graphical Models, Exponential Families, and Variational Inference*, Now Publishers Inc., Hanover, MA, USA.

Wooters, C. & Huijbregts, M. (2008). Multimodal technologies for perception of humans, Springer-Verlag, Berlin, Heidelberg, chapter The ICSI RT07s Speaker Diarization System, pp. 509–519.

Xiang, B., Chaudhari, U. V., Navratil, J., Ramaswamy, G. N. & Gopinath, R. A. (2002). Short-time Gaussianization for robust speaker verification, *Proceedings of ICASSP*, pp. 681–684.

Zhu, X., Barras, C., Meignier, S. & Gauvain, J. (2005). Combining Speaker Identification and BIC for Speaker Diarization, *Proceedings of Interspeech*, pp. 2441 – 2444.

12

Discriminative Universal Background Model Training for Speaker Recognition

Wei-Qiang Zhang and Jia Liu
Department of Electronic Engineering, Tsinghua University
China

1. Introduction

Speaker recognition (SRE), also called as voiceprint recognition, is the problem of determining the identity of the speaker from a sample of speech signal. It is an important branch of speech signal processing and has many potential applications such as in telephone banking, access control, information security, law enforcement and other forensic applications (Bimbot et al., 2004; Campbell Jr., 1997; Cole et al., 1997; Kinnunen & Li, 2010; Reynolds, 2002).

Compared with other biometrics techniques, speaker recognition has its own advantages: (1) It is very convenient, natural and low-cost to acquire the speech sample: it does not need the special devices; the telephone, mobile phone or ordinary microphone is adequate. (2) It can be used remotely: with the ubiquitous telecommunications networks and the Internet, the speech sample can be easily transferred through telephone or VoIP, which makes the remote recognition possible. (3) The speech sample contains many inborn characters: from the speech, we can extract some information about vocal tract, mouth, tongue, soft palate, nasal cavity, and etc. (4) The speech sample also contains some acquired characters, such as tone, volume, pace, rhythm, rhetoric, which reflect speaker's place of living, education level, and some personal habits information.

In speaker recognition, the Gaussian mixture model - universal background model (GMM-UBM) is a classical yet widely used method for text-independent speaker verification (Reynolds et al., 2000). In this method, the target speaker is modeled as a GMM and the imposters are modeled as a UBM. When testing, the speech sample is scored as likelihood by the GMM and UBM respectively, and then the likelihood ratio hypothesis test is used for speaker verification. Besides the GMM-UBM, several other methods are developed recently. The most successful ones include the support vector machine using GMM supper vector (GSV-SVM) (Campbell et al., 2006), which concatenate the GMM mean vectors as the input for SVM training and test, and joint factor analysis (JFA) (Kenny et al., 2007), which jointly models the channel subspace and the speaker subspace. Although other methods achieve rapid progress, GMM-UBM is still the basis for their developments.

As the meanwhile, the discriminative technologies, such as minimum classification error (MCE), maximum mutual information (MMI), minimum phone error (MPE), feature domain MPE (fMPE), have been achieved great success in speech recognition and language recognition (Burget et al., 2006; Juang & Katagiri, 1992; Povey & Kingsbury, 2007; Woodland & Povey, 2002).

In speaker recognition, many discriminative approaches have been reported. As for the GMM-UBM method, the approaches can be divided into two catalogs. (1) Some approaches aim to jointly train the target speaker model and corresponding anti-model. For example, In (Korkmazskiy & Juang, 1996), the MCE criterion is used to adapt talker model (i.e., speaker model) parameters and the corresponding anti-talker model parameters. In (Rosenberg et al., 1998), the minimum verification error (MVE) criterion is used to train the speaker and anti-speaker models and also the decision threshold. In (Ma & Chang, 2003), MMI, MCE, figure of merit (FOM) criteria are used to train the target speaker model and corresponding imposter model. In (Angkititrakul & Hansen, 2007), the training process is divided into two stages: in the first stage, the MCE is used minimize the classification error among the in-set speaker models; in the second stage, the MVE is used to minimize the verification error between the in-set and background models. In (Chao et al., 2008; 2009), the MVE methods are used to reinforce the discriminability between the target speaker model and the target speaker dependent anti-model. (2) Other approaches attempt to discriminatively adapt the target speaker model from the UBM, which can be viewed as the modification of the classical maximum a posteriori (MAP) adaptation (Gauvain & Lee, 1994). For example, In (Zhao et al., 2006), a new speaker adaptation method which combines MAP and reference speaker weighting (RSW) adaptation is presented in a hierarchical multigrained mode. In (Longworth & Gales, 2006), an MMI based adaptation method is reported.

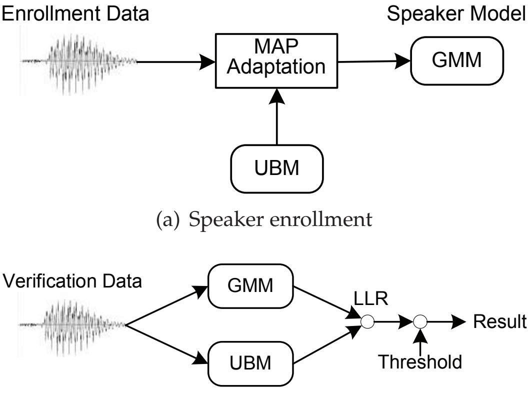
From the discriminative approaches mentioned above, we can find that the UBM is either unchanged or adapted to the target speaker dependent anti-model. If the anti-model is target speaker dependent, it will not be the *universal* background model anymore. But sometimes we have to use the UBM. For example, for fast scoring in GMM-UBM method, we need UBM to determine the orders of mixtures; in the state-of-the-art JFA and GSV-SVM methods, we need UBM to calculate the statistics or the GMM mean vectors. So herein, we want to discriminatively train the UBM to improve its performance.

In order to improve the quality of UBM, many researchers try to select suitable data. For example, in (Hasan et al., 2010; Huang & Li, 2010; Zhang et al., 2010), the data selection based on sub-sampling, maximum entropy and vocal tract length methods are introduced. But as the authors known, there is little report on training the UBM discriminately.

In this chapter, we will discuss the discriminative UBM training method. Firstly we will give a brief review of the GMM-UBM method. After that, we propose our discriminative UBM training method. We will discuss its principle and implementation details. At last, the presented method will be evaluated through large-scale experiments. The results on NIST speaker recognition evaluation dataset will be reported.

2. Overview of GMM-UBM

The GMM-UBM can be viewed as a likelihood-ratio detector: the UBM is trained to represent the speaker-independent distribution of features while the GMM is adapted from the UBM to depict the individual speaker characteristics. In GMM-UBM system, as shown in Fig. 1, a UBM is firstly trained to capture the general characteristics of all the speakers, so it is called *universal* background model. The UBM parameters include weights, mean vectors and covariance matrices, usually denoted by $\boldsymbol{\lambda} = \{w_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}_{m=1}^{M}$, where M is the number of Gaussian mixtures. In speaker recognition, usually the value of M is large, varied from several hundred to several thousand, and the covariance matrices are often set in diagonal form, which facilitates the fast computation.



(a) Speaker enrollment

(b) Speaker verification

Fig. 1. The basic framework of GMM-UBM system

For the t-th frame of feature vector $\boldsymbol{x}_t$, the UBM gives the likelihood as

$$p(\boldsymbol{x}_t|\boldsymbol{\lambda}) = \sum_{m=1}^{M} w_m \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \tag{1}$$

For a T-frame segment $\boldsymbol{x} = \{\boldsymbol{x}_t\}_{t=1}^{T}$, the likelihood is approximated via frame independent assumption as

$$p(\boldsymbol{x}|\boldsymbol{\lambda}) = \prod_{t=1}^{T} p(\boldsymbol{x}_t|\boldsymbol{\lambda}) \tag{2}$$

Usually, the logarithm form of likelihood is used for calculation.

The UBM is usually trained by using the Baum-Welch algorithm (Huang et al., 2000) based on a maximum likelihood (ML) criterion. The Baum-Welch algorithm is in fact a type of expectation-maximization (EM) algorithm and can be implemented iteratively. Suppose the current parameters are obtained, then the new parameters can be updated as

$$w_m^{\text{new}} = \frac{n_m}{T} \tag{3}$$

$$\boldsymbol{\mu}_m^{\text{new}} = \frac{\boldsymbol{f}_m}{n_m} \tag{4}$$

$$\boldsymbol{\Sigma}_m^{\text{new}} = \frac{\boldsymbol{S}_m}{n_m} \tag{5}$$

where n_m, $\boldsymbol{f}_m$ and $\boldsymbol{S}_m$ are the zero-th order, first order and second order statistics

$$n_m = \sum_{t=1}^{T} \gamma_m(\boldsymbol{x}_t) \tag{6}$$

$$\boldsymbol{f}_m = \sum_{t=1}^{T} \gamma_m(\boldsymbol{x}_t)\boldsymbol{x}_t \tag{7}$$

$$\boldsymbol{S}_m = \sum_{t=1}^{T} \gamma_m(\boldsymbol{x}_t)\boldsymbol{x}_t\boldsymbol{x}_t^{\text{T}} \tag{8}$$

$\gamma_m(\boldsymbol{x}_t)$ is m-th mixture of occupation probability

$$\gamma_m(\boldsymbol{x}_t) = \frac{w_m \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{m'=1}^{M} w_{m'} \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_{m'}, \boldsymbol{\Sigma}_{m'})} \tag{9}$$

The initial parameters can be set as: $w_m = 1/M$, $\boldsymbol{\Sigma}_m = \boldsymbol{I}$ and each $\boldsymbol{\mu}_m$ can be randomly selected from the training samples or use the finer Lind-Buzo-Gray (LBG) algorithm to get the initial values. Through enough iterations, the local maximum of the likelihood can be achieved and the parameters become stable.

After the UBM is trained, in the enrollment stage, the mean vectors of UBM is adapted by using enrollment data $\boldsymbol{x}^s$ of speaker s under MAP criterion (Gauvain & Lee, 1994).

$$\boldsymbol{\mu}_m^s = \frac{n_m}{n_m + \gamma} \frac{\boldsymbol{f}_m}{n_m} + \frac{\gamma}{n_m + \gamma} \boldsymbol{\mu}_m \tag{10}$$

where n_m and $\boldsymbol{f}_m$ are calculated by using enrollment segment $\boldsymbol{x}^s$, γ is the relevance factor, and usually set as 16 (Reynolds et al., 2000). Note that, the weights and covariance matrices are not updated. Thus, the parameters for GMM of speaker s are $\boldsymbol{\lambda}^s = \{w_m, \boldsymbol{\mu}_m^s, \boldsymbol{\Sigma}_m\}_{m=1}^{M}$.

In the speaker verification stage, the log-likelihood-ratio (LLR) of the test segment $\boldsymbol{x}^r$ is calculated by using the GMM and the UBM, and compared with threshold to give the last acceptance or rejection decision.

$$s(\boldsymbol{x}^r, \boldsymbol{\lambda}^s, \boldsymbol{\lambda}) = \frac{1}{T_r}(\log p(\boldsymbol{x}^r|\boldsymbol{\lambda}^s) - \log p(\boldsymbol{x}^r|\boldsymbol{\lambda})) \gtrless s_{\text{th}} \tag{11}$$

where T_r is the number of frames of verification segment $\boldsymbol{x}^r$. This equation can be expanded as

$$s(\boldsymbol{x}^r, \boldsymbol{\lambda}^s, \boldsymbol{\lambda}) = \frac{1}{T_r} \sum_{t=1}^{T_r} (\log \sum_{m=1}^{M} w_m^s \mathcal{N}(\boldsymbol{x}_t^r; \boldsymbol{\mu}_m^s, \boldsymbol{\Sigma}_m^s) - \log \sum_{m=1}^{M} w_m \mathcal{N}(\boldsymbol{x}_t^r; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)) \tag{12}$$

Note that in our case, $w_m^s = w_m$, $\boldsymbol{\Sigma}_m^s = \boldsymbol{\Sigma}_m$, and $\boldsymbol{\mu}_m^s$ is adapted from $\boldsymbol{\mu}_m$. This means that the scores calculated by the corresponding mixtures of GMM and UBM are approximately equal. According to the property of GMM, we know that each feature frame is located at a local region, that is to say, most mixtures will give very small scores for each frame. So we can neglect these mixtures and only calculate top N mixtures for LLR scoring.

$$s(\boldsymbol{x}^r, \boldsymbol{\lambda}^s, \boldsymbol{\lambda}) = \frac{1}{T_r} \sum_{t=1}^{T_r} (\log \sum_{n=1}^{N} w_{m_n(t)}^s \mathcal{N}(\boldsymbol{x}_t^r; \boldsymbol{\mu}_{m_n(t)}^s, \boldsymbol{\Sigma}_{m_n(t)}^s) - \log \sum_{n=1}^{N} w_{m_n(t)} \mathcal{N}(\boldsymbol{x}_t^r; \boldsymbol{\mu}_{m_n(t)}, \boldsymbol{\Sigma}_{m_n(t)})) \tag{13}$$

where $\{m_n(t)\}_{n=1}^{N}$ are the top N scoring mixture indices calculated by UBM for the frame $\boldsymbol{x}_t$. This fast scoring strategy is introduced in (Reynolds et al., 2000) and widely used in GMM-UBM method and other similar circumstances.

3. Discriminative UBM training

From the above section, we can see that the UBM is trained under ML criterion. This criterion is asymptotically optimal, in another word, it is optimal if there are infinite amount of training data. In practice, this condition can not be satisfied. The available training data is always limited. As a consequence, likelihood based training can not guarantee optimal performance.

For speaker verification systems, the most important performance measure is the verification errors. So we borrow the minimum verification error (MVE) criterion (Rosenberg et al., 1998) to develop a discriminative UBM training method.

Note that our motivation is different to other discriminative approaches for speaker recognition: we only want to obtain a high quality UBM. The flowchart is showed in Fig. 2. We can observed that, the enrollment data and verification data are all our *training* data.

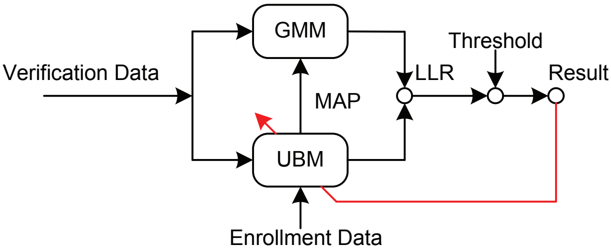


Fig. 2. The flowchart for discriminative UBM training

3.1 Discriminative framework

Similar to MCE criterion (Juang & Katagiri, 1992), the MVE criterion also can optimized by using generalized probabilistic descent (GPD) framework. To implement it, a *smoothed* loss function should be defined first and then the gradient descent method is used to obtain the (local) minimum of the loss function.

Firstly, we define the false verification function (similar to discriminant function in MCE) as

$$d(i, \boldsymbol{\lambda}) = [\log p(\boldsymbol{x}^r|\boldsymbol{\lambda}^s) - \log p(\boldsymbol{x}^r|\boldsymbol{\lambda}) - s_{\text{th}}]\,\text{sign}(i) \tag{14}$$

where i denotes the i-th trial which involves s-th speaker model and r-th verification segment, and

$$\text{sign}(i) = \begin{cases} -1 & \text{if } i \text{ is target trial} \\ 1 & \text{if } i \text{ is non-target trial} \end{cases} \tag{15}$$

From (14), we can see that $d(i, \boldsymbol{\lambda}) > 0$ indicates trial i is a false verification and $d(i, \boldsymbol{\lambda}) < 0$ implies a correct verification. The value of the false verification function indicates the distortion between the models and the corresponding training data. The larger the false verification function is, the more adjustment of the model parameters is required to improve the verification performance.

Next, we will define the loss function. In general, the loss function is a function of the false verification function. Obviously, the loss function and the false verification function can be defined individually. Loss function is used to show the cost of mis-verification a trial. It is required that the loss function should be a differentiable, and monotonically non-decreasing function. Usually, sigmoid function is a good choice. The gradients of this function are easy to be obtained. The loss function is defined as

$$l(i, \boldsymbol{\lambda}) = \frac{\text{cost}(i)}{1 + \exp\{-\alpha d(i, \boldsymbol{\lambda})\}} \tag{16}$$

where α is the slop parameter of sigmoid function. cost(i) is the cost of false verification of i-th trial.

Then, the objective function (total loss) need to be minimized is

$$L(\boldsymbol{\lambda}) = \sum_{i=1}^{I} l(i,\boldsymbol{\lambda})u(l(i,\boldsymbol{\lambda}) + \delta) \tag{17}$$

where $u(\cdot)$ is a unit function

$$u(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if x < 0} \end{cases} \tag{18}$$

and δ is a small positive number.
From (17), it is clear that the incorrectly verified trials (for the trials such that $l(i,\boldsymbol{\lambda}) > 0$) and the correctly verified but near the decision boundary trials (for the trials such that $0 \geq l(i,\boldsymbol{\lambda}) \geq -\delta$) are used for training.
We can use the gradient descent algorithm to optimize this objective function. Note that herein we only discuss the mean vectors. Other parameters can be obtained similarly. The update formula is

$$\boldsymbol{\mu}_m(n+1) = \boldsymbol{\mu}_m(n) - \varepsilon_n \frac{\partial L(\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} \tag{19}$$

where ε_n is the step factor.
In practise, we can use Baum-Welch algorithm to obtain the parameters of UBM initially, then use (19) to update its mean vectors discriminatively.

3.2 Gradients

For the gradient descent algorithm, the most important step is to obtain the gradients of the objective function. It is not easy but straightforward. We will solve this problem step by step. The gradient of the objective function w.r.t the mean vector is

$$\begin{aligned} \frac{\partial L(\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} &= \sum_{i=1}^{I} \frac{\partial l(i,\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} \\ &= \sum_{i=1}^{I} \frac{\partial l(i,\boldsymbol{\lambda})}{\partial d(i)} \frac{\partial d(i,\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} \\ &= \sum_{i=1}^{I} \frac{\alpha}{\text{cost}(i)} l(i,\boldsymbol{\lambda}) \left[\text{cost}(i) - l(i,\boldsymbol{\lambda})\right] \frac{\partial s(i,\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} \text{sign}(i) \end{aligned} \tag{20}$$

where $\partial s(i,\boldsymbol{\lambda})/\partial \boldsymbol{\mu}_m$ consists of two items

$$\frac{\partial s(i,\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} = \frac{1}{T_r} \left(\frac{\partial \log p(\boldsymbol{x}^r|\boldsymbol{\lambda}^s)}{\partial \boldsymbol{\mu}_m} - \frac{\partial \log p(\boldsymbol{x}^r|\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} \right) \tag{21}$$

For the first item, because

$$\begin{aligned} \log p(\boldsymbol{x}^r|\boldsymbol{\lambda}) &= \sum_{t=1}^{T_r} \log p(\boldsymbol{x}_t^r|\boldsymbol{\lambda}) \\ &= \sum_{t=1}^{T_r} \log \sum_{m=1}^{M} w_m \mathcal{N}(\boldsymbol{x}_t^r; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \end{aligned} \tag{22}$$

thus, we can obtain

$$\begin{aligned}\frac{\partial \log p(\boldsymbol{x}^r|\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} &= \sum_{t=1}^{T_r} \frac{1}{p(\boldsymbol{x}_t^r|\boldsymbol{\lambda})} \frac{\partial w_m \mathcal{N}(\boldsymbol{x}_t^r; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\partial \boldsymbol{\mu}_m} \\ &= \sum_{t=1}^{T_r} -2\gamma_m(\boldsymbol{x}_t^r)\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}_t^r - \boldsymbol{\mu}_m) \end{aligned} \tag{23}$$

For the second item, according to (10), we know that $\{\boldsymbol{\mu}_m^s\}_{m=1}^M$ is a function of $\{\boldsymbol{\mu}_m\}_{m=1}^M$. Based on the chain rule for derivation, we have

$$\frac{\partial \log p(\boldsymbol{x}_t^r|\boldsymbol{\lambda}^s)}{\partial \boldsymbol{\mu}_m} = \sum_{t=1}^{T_r} \frac{1}{p(\boldsymbol{x}^r|\boldsymbol{\lambda}^s)} \sum_{m'=1}^{M} \frac{\partial (\boldsymbol{\mu}_{m'}^s)^{\mathrm{T}}}{\partial \boldsymbol{\mu}_m} \frac{\partial w_{m'}^s \mathcal{N}(\boldsymbol{x}_t^r; \boldsymbol{\mu}_{m'}^s, \boldsymbol{\Sigma}_{m'}^s)}{\partial \boldsymbol{\mu}_{m'}^s} \tag{24}$$

Similar to (23), we can obtain

$$\frac{1}{p(\boldsymbol{x}_t^r|\boldsymbol{\lambda}^s)} \frac{\partial w_{m'}^s \mathcal{N}(\boldsymbol{x}_t^r; \boldsymbol{\mu}_{m'}^s, \boldsymbol{\Sigma}_{m'}^s)}{\partial \boldsymbol{\mu}_{m'}^s} = -2\gamma_{m'}^s(\boldsymbol{x}_t^r)(\boldsymbol{\Sigma}_{m'}^s)^{-1}(\boldsymbol{x}_t^r - \boldsymbol{\mu}_{m'}^s) \tag{25}$$

where $\gamma_{m'}^s(\boldsymbol{x}_t^r)$ is the m'-th mixture occupation of $\boldsymbol{x}_t^r$ calculated by GMM of speaker s. Substitute (25) to (24), we get

$$\frac{\partial \log p(\boldsymbol{x}_t^r|\boldsymbol{\lambda}^s)}{\partial \boldsymbol{\mu}_m} = \sum_{t=1}^{T_r} \sum_{m'=1}^{M} -2\gamma_{m'}^s(\boldsymbol{x}_t^r) \frac{\partial (\boldsymbol{\mu}_{m'}^s)^{\mathrm{T}}}{\partial \boldsymbol{\mu}_m} (\boldsymbol{\Sigma}_{m'}^s)^{-1}(\boldsymbol{x}_t^r - \boldsymbol{\mu}_{m'}^s) \tag{26}$$

Next, we will get $\partial(\boldsymbol{\mu}_{m'}^s)^{\mathrm{T}}/\partial \boldsymbol{\mu}_m$. This can be divided into two cases. When $m' = m$

$$\begin{aligned}\frac{\partial (\boldsymbol{\mu}_m^s)^{\mathrm{T}}}{\partial \boldsymbol{\mu}_m} &= \frac{(\sum_{t=1}^{T_s} \frac{\partial \gamma_m(\boldsymbol{x}_t^s)}{\partial \boldsymbol{\mu}_m}(\boldsymbol{x}_t^s)^{\mathrm{T}} + \gamma \boldsymbol{I})(n_m + \gamma) - \frac{\partial n_m}{\partial \boldsymbol{\mu}_m}(\sum_{t=1}^{T_s} \gamma_m(\boldsymbol{x}_t^s)\boldsymbol{x}_t^s + \gamma \boldsymbol{\mu}_m)^{\mathrm{T}}}{(n_m + \gamma)^2} \\ &= \frac{\sum_{t=1}^{T_s} \frac{\partial \gamma_m(\boldsymbol{x}_t^s)}{\partial \boldsymbol{\mu}_m}(\boldsymbol{x}_t^s)^{\mathrm{T}} + \gamma \boldsymbol{I}}{n_m + \gamma} - \frac{\partial n_m}{\partial \boldsymbol{\mu}_m} \frac{(\boldsymbol{\mu}_m^s)^{\mathrm{T}}}{n_m + \gamma} \\ &= \frac{\sum_{t=1}^{T_s} \frac{\partial \gamma_m(\boldsymbol{x}_t^s)}{\partial \boldsymbol{\mu}_m}(\boldsymbol{x}_t^s - \boldsymbol{\mu}_m^s)^{\mathrm{T}} + \gamma \boldsymbol{I}}{n_m + \gamma} \end{aligned} \tag{27}$$

where T_s is the number of frames of enrollment segment $\boldsymbol{x}^s$ and

$$\frac{\partial \gamma_m(\boldsymbol{x}_t^s)}{\partial \boldsymbol{\mu}_m} = 2(\gamma_m(\boldsymbol{x}_t^s) - \gamma_m^2(\boldsymbol{x}_t^s))\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}_t^s - \boldsymbol{\mu}_m) \tag{28}$$

When $m' \neq m$

$$\begin{aligned}\frac{\partial (\boldsymbol{\mu}_{m'}^s)^{\mathrm{T}}}{\partial \boldsymbol{\mu}_m} &= \frac{(\sum_{t=1}^{T_s} \frac{\partial \gamma_{m'}(\boldsymbol{x}_t^s)}{\partial \boldsymbol{\mu}_m}(\boldsymbol{x}_t^s)^{\mathrm{T}})(n_{m'} + \gamma) - \frac{\partial n_{m'}}{\partial \boldsymbol{\mu}_m}(\sum_{t=1}^{T_s} \gamma_{m'}(\boldsymbol{x}_t^s)\boldsymbol{x}_t^s + \gamma \boldsymbol{\mu}_{m'})^{\mathrm{T}}}{(n_{m'} + \gamma)^2} \\ &= \frac{\sum_{t=1}^{T_s} \frac{\partial \gamma_{m'}(\boldsymbol{x}_t^s)}{\partial \boldsymbol{\mu}_m}(\boldsymbol{x}_t^s)^{\mathrm{T}}}{n_{m'} + \gamma} - \frac{\partial n_{m'}}{\partial \boldsymbol{\mu}_m} \frac{(\boldsymbol{\mu}_{m'}^s)^{\mathrm{T}}}{n_{m'} + \gamma} \\ &= \frac{\sum_{t=1}^{T_s} \frac{\partial \gamma_{m'}(\boldsymbol{x}_t^s)}{\partial \boldsymbol{\mu}_m}(\boldsymbol{x}_t^s - \boldsymbol{\mu}_{m'}^s)^{\mathrm{T}}}{n_{m'} + \gamma} \end{aligned} \tag{29}$$

where

$$\frac{\partial \gamma_{m'}(\boldsymbol{x}_t^s)}{\partial \boldsymbol{\mu}_m} = 2\gamma_{m'}(\boldsymbol{x}_t^s)\gamma_m(\boldsymbol{x}_t^s)\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}_t^s - \boldsymbol{\mu}_m) \tag{30}$$

Until now, we have obtained all the gradients through manual derivation. The computation of these gradients are not easy to implement, so we only consider the diagonal elements of $\partial(\boldsymbol{\mu}_{m'}^s)^{\mathrm{T}}/\partial\boldsymbol{\mu}_m$. We define

$$\boldsymbol{D} = \mathrm{diag}\left\{ \frac{\sum_{t=1}^{T_s} 2(\gamma_m(\boldsymbol{x}_t^s) - \gamma_m^2(\boldsymbol{x}_t^s))\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}_t^s - \boldsymbol{\mu}_m)(\boldsymbol{x}_t^s - \boldsymbol{\mu}_m^s)^{\mathrm{T}} + \gamma \boldsymbol{I}}{n_m + \gamma} \right\} \tag{31}$$

Using this diagonal matrix, (21) will become

$$\frac{\partial s(i, \boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} = -\frac{2}{T_r}\left\{ \sum_{t=1}^{T_r} \left[\gamma_m^s(\boldsymbol{x}_t^r)\boldsymbol{D}(\boldsymbol{\Sigma}_m^s)^{-1}(\boldsymbol{x}_t^r - \boldsymbol{\mu}_m^s) - \gamma_m(\boldsymbol{x}_t^r)\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}_t^r - \boldsymbol{\mu}_m) \right] \right\} \tag{32}$$

Substitute (32) to (20), we can get the simplified version of gradients.

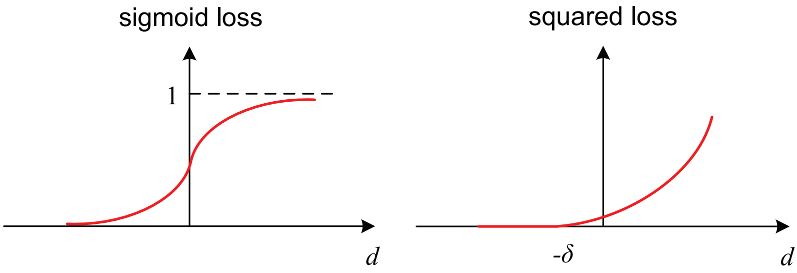
4. Squared loss function

In the gradient-type descent algorithms, the loss function decrease as the false verification function decreases. However, if the loss function is defined improperly, the verification performance will not be improved through discriminative training.

Besides the sigmoid loss function, the squared loss function (Chao et al., 2008) is also used. It can be expressed as

$$l(i, \boldsymbol{\lambda}) = \begin{cases} \mathrm{cost}(i)\alpha(d(i, \boldsymbol{\lambda}) + \delta)^2 & \text{if } d(i, \boldsymbol{\lambda}) \geq \delta \\ 0 & \text{otherwise} \end{cases} \tag{33}$$

where α and δ are control parameters. Unlike sigmoid function, the squared loss function has greater gradient for large d, which gives more penalty for the severe false verification segments. To give an intuitive illustration, we borrow a figure from (Chao et al., 2009) and show it in Fig. 3.



(a) sigmoid loss function (b) squared loss function

Fig. 3. Comparison of sigmoid loss function and squared loss function (Chao et al., 2009)

Using this squared loss function, the gradient of the objective function w.r.t the mean vector will be

$$\frac{\partial L(\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} = \sum_{i=1}^{I} 2\alpha \text{cost}(i)(d(i,\boldsymbol{\lambda}) + \delta)\frac{\partial s(i,\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m}\text{sign}(i) \tag{34}$$

Other derivations are the same as that in Section 3.

5. Approximate conjugate gradient algorithm

To decrease the object function, gradient descent algorithms in Section 3. In fact, in optimization, other methods, such as conjugate gradient algorithm, are usually used. The gradient descent algorithm is simple to implement, since it only requires first-order derivatives. But its convergent rate is slow. In contrast, the conjugate gradient algorithm has good convergent property, but unfortunately it requires second-order derivatives. In Section 3, we can see that the first-order derivatives are very difficult to deal with, not to mention the second-order derivatives. Herein, we introduce another optimization method, namely, approximate conjugate gradient algorithm (Dixon, 1972), which only needs the first-order derivatives but with fast convergent rate. For convenient expressing, we first define

$$\boldsymbol{g} = \frac{\partial L(\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} \tag{35}$$

By using the approximate conjugate gradient algorithm, the update formula will be

$$\boldsymbol{\mu}_m(n+1) = \boldsymbol{\mu}_m(n) - \varepsilon_n \boldsymbol{p}_n \tag{36}$$

where ε_n is the step factor and $\boldsymbol{p}_n$ can be viewed as modified gradient, which can be expressed as

$$\boldsymbol{p}_n = \boldsymbol{g}_n - \beta \boldsymbol{p}_{n-1} \tag{37}$$

where

$$\beta = \frac{\boldsymbol{g}_n^{\mathrm{T}}(\boldsymbol{g}_n - \boldsymbol{g}_{n-1})}{\|\boldsymbol{g}_n\|_2^2} \tag{38}$$

and $\|\cdot\|_2^2$ is the squared 2-norm.

6. Experimental results

6.1 Experimental setup

In this section, the experiments are carried out on NIST speaker recognition evaluation corpora (NIST, 2010). The UBM training (i.e., ML traing) data are selected from SRE04 1-side training set. The discriminative UBM training data come from SRE05 core test condition (1conv4w-1conv4w) dataset. The test data come from SRE06 core test condition (1conv4w-1conv4w) dataset. The numbers of trials of SRE05 and SRE06 are summarized in Table 1.

For the frontend, speech/silence segmentation is performed by a G.723.1 VAD detector. 12 MFCC coefficients plus C0 are computed using 20 ms window and 10ms shift. Cepstral mean subtraction and feature warping (Pelecanos & Sridharan, 2001) with a 3 s window are

Dataset	Target trial	Non-target trial
SRE05 female	1540	16238
SRE05 male	1226	12398
SRE06 female	2712	27913
SRE06 male	2061	21211

Table 1. NIST SRE05 and SRE06 1conv4w-1conv4w trial summary

applied for channel mismatch compensation. Delta, acceleration and triple-delta coefficients are appended to each feature vector, which results in a dimensionality of 52. After that, 25% of low energy frames are discarded using a dynamic threshold. Then, HLDA is employed to decorrelate features and reduce the dimensionality from 52 to 39. Finally, a feature domain latent factor analysis (fLFA) (Vair et al., 2006) is applied to compensate the channel distortion. The performance measures are the same as NIST speaker recognition evaluation (NIST, 2010), using equal error rate (EER) and minimum detection cost function (DCF). DCF is defined as

$$\mathrm{DCF} = 0.1P_{\mathrm{miss}} + 0.99P_{\mathrm{fa}} \tag{39}$$

where P_{miss} is the miss probability and P_{fa} is false alarm probability. We vary the decision threshold, the EER is achieved when P_{miss} is equal to P_{fa}; the min DCF is achieved when DCF get its minimum.

6.2 Baseline performance

A GMM-UBM system has been built as baseline for contrastive analysis. The gender-dependant UBMs with 256 mixtures are trained. No score normalization technology is used.

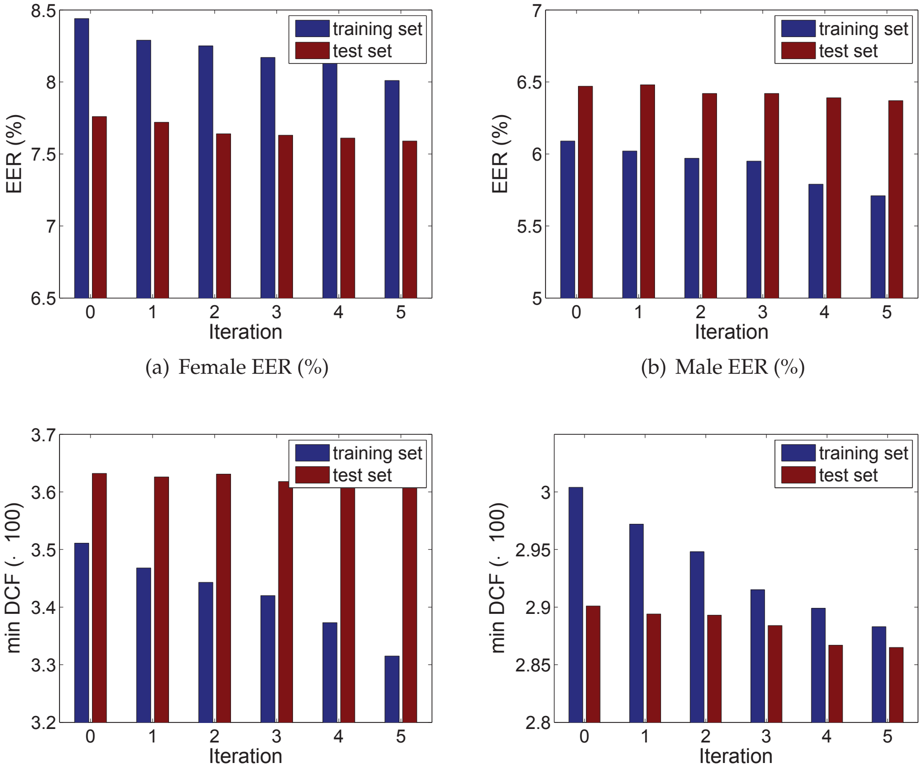
The performance of GMM-UBM system on SRE06 dataset is listed in Table 2. The EERs for female is 7.76% and for male is 6.47%. For 256-mixture GMM-UBM system, this is a quite good baseline.

Gender	EER (%)	min DCF ($\times$100)
female	7.76	3.63
male	6.47	2.90

Table 2. Performance of baseline GMM-UBM system

6.3 Sigmoid loss function

In this section, discriminative UBM training with sigmoid loss function is tested. We use SRE05 as training set and SRE06 as test set. The performance on training set and test set are both given in Fig. 4, and the results on test set are listed in Table 3. We can see that after discriminative UBM training, the EERs and min DCFs for female and male are all decreased slightly. This shows that the discriminative UBM training is better than the generative training.



(a) Female EER (%)

(b) Male EER (%)

(c) Female min DCF (× 100)

(d) Male min DCF (× 100)

Fig. 4. Performance of discriminative UBM training with sigmoid loss function

Gender	EER (%)	min DCF (×100)
female	7.59	3.61
male	6.37	2.87

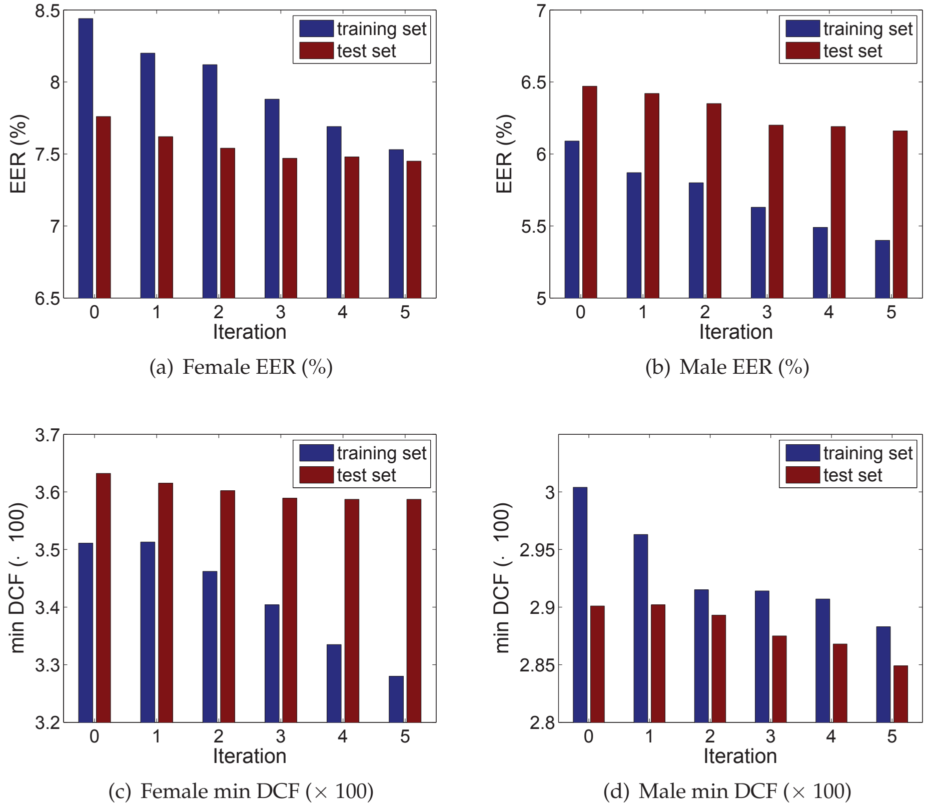
Table 3. Performance of discriminative UBM training with sigmoid loss function

6.4 Squared loss function

In this section, we change the sigmoid loss function to squared loss function. The performance on training set and test set are both given in Fig. 5, and the results on test set are listed in Table 4. Compared these results with that in Section 6.3, it can be observed that the squared loss function is better than the sigmoid loss function. This is due to the more penalty for the falser verification segments.

6.5 Approximate conjugate gradient algorithm

In this section, we change the gradient descent algorithm to approximate conjugate gradient algorithm. The EERs and min DCFs are showed in Fig. 6 and Table 5. We can see that



(a) Female EER (%)

(b) Male EER (%)

(c) Female min DCF (× 100)

(d) Male min DCF (× 100)

Fig. 5. Performance of discriminative UBM training with squared loss function

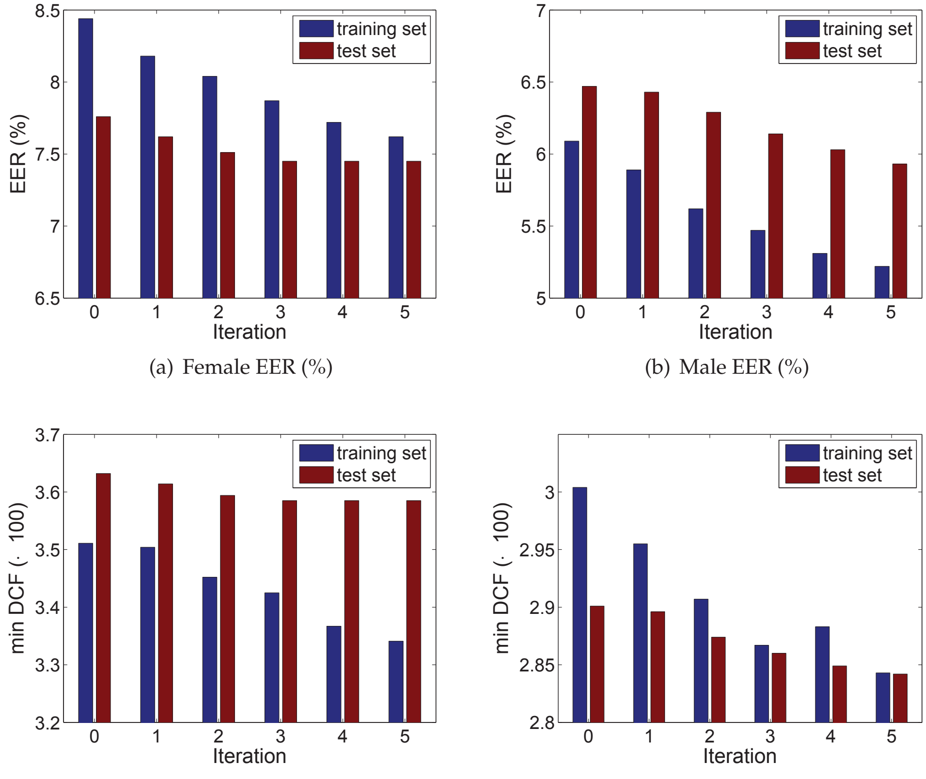
Gender	EER (%)	min DCF (×100)
female	7.45	3.59
male	6.16	2.85

Table 4. Performance of discriminative UBM training with squared loss function

the last female performance of approximate conjugate gradient algorithm is similar to that of gradient descent algorithm, but with faster convergence speed. For the male gender, the approximate conjugate gradient algorithm is better than the gradient descent algorithm. This shows the effectiveness of the approximate conjugate gradient algorithm. At last, we compare the detection error tradeoff (DET) curves (Martin et al., 1997) of the the baseline system and discriminative UBM training with approximate conjugate gradient algorithm in Fig. 7. In the figures, The circles denote the min DCF operating points. From the DET curves, we can see that our proposed discriminative UBM training method achieves slightly better performance.

Gender	EER (%)	min DCF (×100)
female	7.45	3.59
male	5.93	2.84

Table 5. Performance of discriminative UBM training with approximate conjugate gradient algorithm



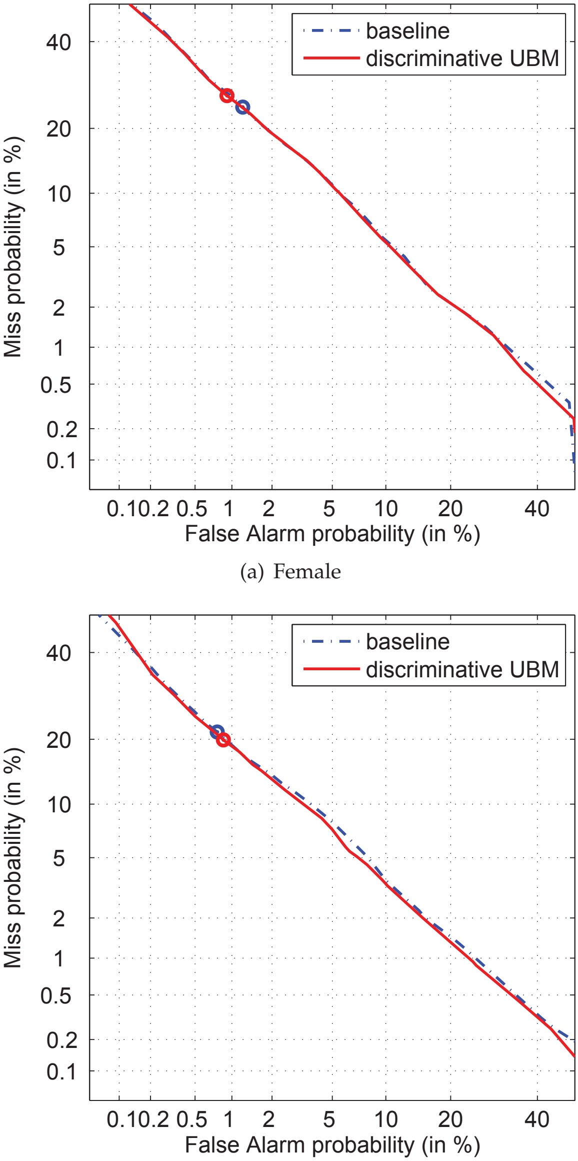
(a) Female EER (%)

(b) Male EER (%)

(c) Female min DCF (× 100)

(d) Male min DCF (× 100)

Fig. 6. Performance of discriminative UBM training with approximate conjugate gradient algorithm



(a) Female

(b) Male

Fig. 7. The DET curves of baseline system and discriminative UBM system

7. Conclusion

In this chapter, we present a discriminative UBM training method for speaker recognition. We build the discriminative framework and derive the update formula under minimum verification error criterion. In this framework, we compare the sigmoid loss function and squared loss function, the gradient descent algorithm and the approximate conjugate gradient algorithm. The experimental results show that the our proposed discriminative UBM training method is better than the prevalent ML training method.

8. Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No. 61005019, No. 90920302 and No. 60931160443, and in part by the National High Technology Development Program of China under Grant No. 2008AA040201.

9. References

Angkititrakul, P. & Hansen, J. H. L. (2007). Discriminative in-set/out-of-set speaker recognition, *IEEE Transactions on Audio, Speech and Language Processing* 15(2): 498 – 508.

Bimbot, F., Bonastre, J.-F., Fredouille, C., Gravier, G., Magrin-Chagnolleau, I., Meignier, S., Merlin, T., Ortega-Garcia, J., Petrovska-Delacretaz, D. & Reynolds, D. A. (2004). A tutorial on text-independent speaker verification, *EURASIP Journal on Applied Signal Processing* 2004(4): 430–451.

Burget, L., Matejka, P. & Cernocky, J. (2006). Discriminative training techniques for acoustic language identification, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1, Toulouse, pp. 209–212.

Campbell Jr., J. P. (1997). Speaker recognition: A tutorial, *Proceedings of the IEEE* 85(9): 1437–1462.

Campbell, W., Sturim, D. & Reynolds, D. (2006). Support vector machines using GMM supervectors for speaker verification, *IEEE Signal Processing Letters* 13(5): 308 – 311.

Chao, Y.-H., Tsai, W.-H. & Wang, H.-M. (2008). Discriminative feedback adaptation for GMM-UBM speaker verification, *Proc. International Symposium on Chinese Spoken Language Processing*, Kunming, pp. 169–172.

Chao, Y.-H., Tsai, W.-H. & Wang, H.-M. (2009). Improving GMM-UBM speaker verification using discriminative feedback adaptation, *Computer Speech and Language* 23(3): 376–388.

Cole, R. A., Mariani, J., Uszkoreit, H. et al. (1997). *Survey of the State of the Art in Human Language Technology*, The Press Syndicate of the University of Cambridge, New York.

Dixon, L. C. W. (1972). *Nonlinear Optimisation*, The English University Press, London.

Gauvain, J.-L. & Lee, C.-H. (1994). Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains, *IEEE Transactions on Speech and Audio Processing* 2(2): 291 – 298.

Hasan, T., Lei, Y., Chandrasekaran, A. & Hansen, J. H. L. (2010). A novel feature sub-sampling method for efficient universal background model training in speaker verification, *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Dallas, pp. 4494 – 4497.

Huang, C.-L. & Li, H. (2010). UBM data selection for effective speaker modeling, *Proc. International Symposium on Chinese Spoken Language Processing*, Taiwan, pp. 162–165.

Huang, X.-D., Acero, A. & Hon, H.-W. (2000). *Spoken Language Processing*, Prentice Hall, New Jersey.

Juang, B.-H. & Katagiri, S. (1992). Discriminative learning for minimum error classification, *IEEE Transactions on Signal Processing* 40(12): 3043 – 3054.

Kenny, P., Boulianne, G., Ouellet, P. & Dumouchel, P. (2007). Joint factor analysis versus eigenchannels in speaker recognition, *IEEE Transactions on Audio, Speech and Language Processing* 15(4): 1435 – 1447.

Kinnunen, T. & Li, H. (2010). An overview of text-independent speaker recognition: From features to supervectors, *Speech Communication* 52(1): 12 – 40.

Korkmazskiy, F. & Juang, B.-H. (1996). Discriminative adaptation for speaker verification, *Proc. International Conference on Spoken Language Processing*, Vol. 3, Philadelphia, pp. 1744–1747.

Longworth, C. & Gales, M. (2006). Discriminative adaptation for speaker verification, *Proc. InterSpeech 2006 and 9th International Conference on Spoken Language Processing*, Vol. 3, Pittsburgh, pp. 1467–1470.

Ma, C. & Chang, E. (2003). Comparison of discriminative training methods for speaker verification, *Proc. IEEE International Conference on Acoustic, Speech and Signal Processing*, Vol. 1, Hong Kong, pp. 192–195.

Martin, A., Doddington, G., Kamm, T., Ordowski, M. & Przybocki, M. (1997). The DET curve in assessment of detection task performance, *Proc. European Conference on Speech Communication and Technology*, Rhodes, pp. 1895–1898.

NIST (2010). NIST Speaker Recognition Evaluation, [Online], Available: http://www.itl.nist.gov/iad/mig/tests/sre.

Pelecanos, J. & Sridharan, S. (2001). Feature warping for robust speaker verification, *Proc. IEEE Odyssey - The Speaker and Language Recognition Workshop*, Crete, pp. 213–218.

Povey, D. & Kingsbury, B. (2007). Evaluation of proposed modifications to MPE for large scale discriminative training, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 4, Honolulu, pp. 321–324.

Reynolds, D. A. (2002). An overview of automatic speaker recognition technology, *Proc. IEEE International Conference on Acoustic, Speech and Signal Processing*, Vol. 4, Orlando, pp. 4072–407.

Reynolds, D. A., Quatieri, T. F. & Dunn, R. B. (2000). Speaker verification using adapted gaussian mixture models, *Digital Signal Processing* 10(1-3): 19–41.

Rosenberg, A. E., Siohan, O. & Parthasarathy, S. (1998). Speaker verification using minimum verification error training, *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, Seattler, pp. 105–108.

Vair, C., Colibro, D., Castaldo, F. et al. (2006). Channel factors compensation in model and feature domain for speaker recognition, *Proc. IEEE Odyssey - The Speaker and Language Recognition Workshop*, San Juan.

Woodland, P. C. & Povey, D. (2002). Large scale discriminative training of hidden Markov models for speech recognition, *Computer Speech and Language* 16(1): 25–47.

Zhang, W.-Q., Shan, Y. & Liu, J. (2010). Multiple background models for speaker verification, *Proc. Odyssey - The Speaker and Language Recognition Workshop*, Brno, pp. 47–51.

Zhao, X., Dong, Y., Luo, J., Yang, H. & Wang, H. (2006). Multigrained model adaptation with MAP and reference speaker weighting for text independent speaker verification, *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, Toulouse, pp. 913–916.

Part 6

Applications

13

Building a Visual Front-end for Audio-Visual Automatic Speech Recognition in Vehicle Environments

Robert Hursig and Jane Zhang
California Polytechnic State University, USA

1. Introduction

Automatic speech recognition (ASR) holds the promise of providing a natural, efficient, and safer means for communication between humans and computers and can profoundly change the way we live. Since its invention in the 1950s, ASR has witnessed considerable research activities and in recent years is finding its way into practical applications as evidenced by more and more consumer devices such as PDAs and mobile phones adding ASR features. While mainstream ASR has focused almost exclusively on the acoustic signal, the performance of these systems degrades considerably in the real-world in the presence of noise. One way to overcome this limitation is to supplement the acoustic speech with a visual signal that remains unaffected in an audibly noisy environment, yielding what is known as audio-visual automatic speech recognition (AVASR).

While previous research demonstrated that the visual modality is a viable tool for identifying speech [1-4], the visual information has yet to become utilized in mainstream AVASR. Despite years of research attention, there has been limited success in creating a system that can reliably detect lips in unconstrained imagery. Existing systems employ methods such as snake and active shape models [5,6], Markov Random Field (MRF) techniques [7], and multi-class, shape-guided fuzzy c-means (FCM) clustering algorithm [8], to detect and locate lips within an image. While the results are commendable, the extensive calculations demanded by these methods are significant. Moreover, a majority of existing lip localization techniques focused on lip parameter extraction within controlled environments with ample image resolution. Within the unconstrained visual environment, AVASR systems must compete with constantly changing lighting conditions and background clutter as well as subject movement in three dimensions. The difficulty of accurately and reliably detecting and tracking lips in unconstrained imagery is a major obstacle in the development of a practical AVASR system in the real world.

In this work we directly address the unconstrained imagery in the development of the visual front end of a practical AVASR system. Generally, the in-car audio-visual environment can be considered as a worst-case scenario for AVASR. Background noise and mechanical vibrations from traveling vehicles severely decreases operational signal-to-noise ratios for audio processing. Several products such as Ford Motor Company’s Sync® and BMW’s high-end Voice Command System use strictly audio information to recognize user requests. However these systems notably suffer from user voice dependence and background noise such as open windows or ambient noise from highway speeds. Likewise, the visual environment inside a car is also challenging, imposing rapidly changing lighting conditions, moving faces within the vehicle, and constantly changing background clutter. In this work, algorithms were developed based on training and test datasets drawn from the AVICAR database [9] that was collected in such an environment. This database contains audio-visual recordings of 50 male and 50 female participants with varying ethnicities, constantly changing lighting conditions and cluttered background within a moving automobile. Video and image resolution for this database is 240-by-360 pixels, height-by-width.

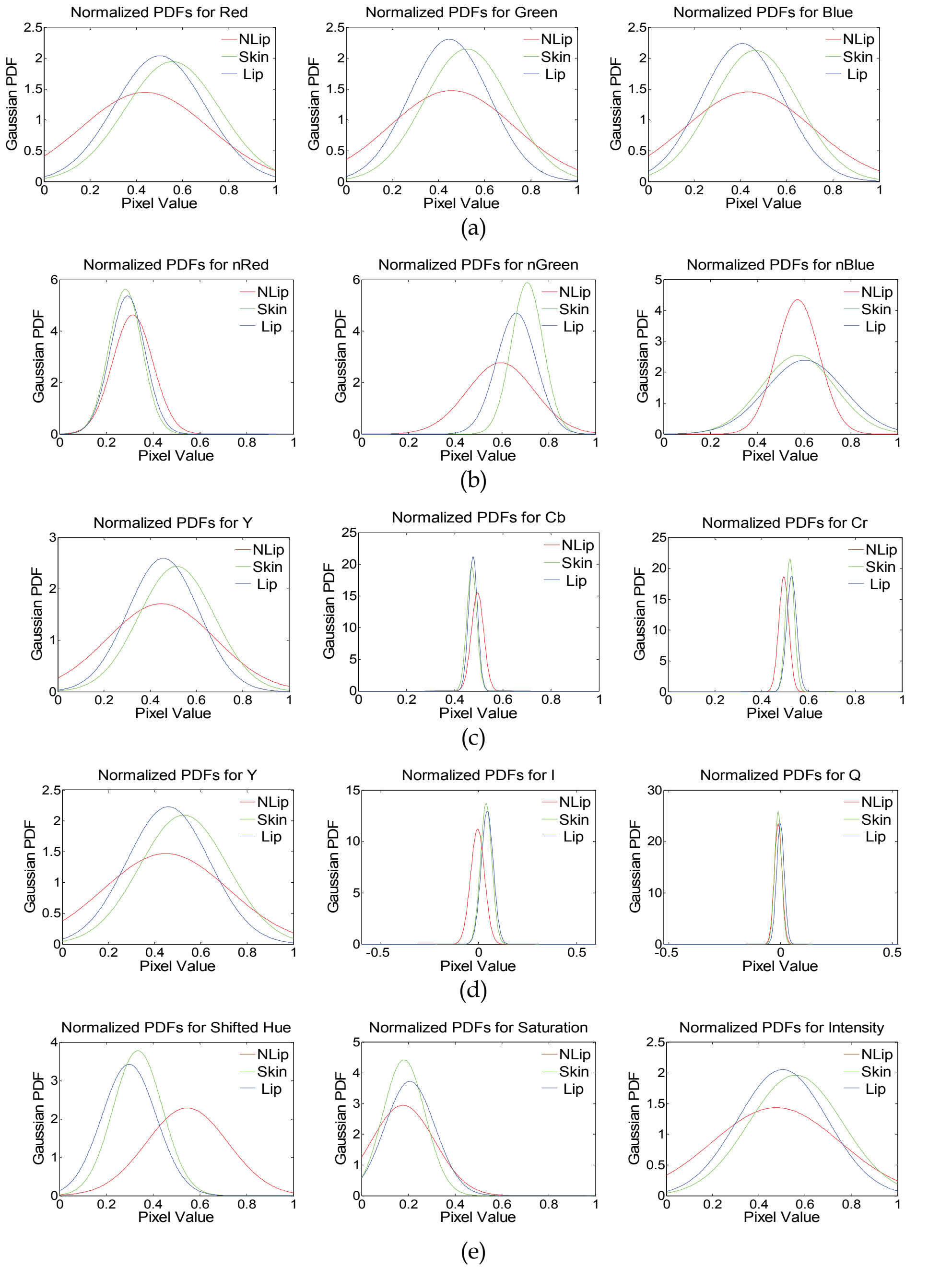
The goal of this work is to develop a robust image lip localization algorithm designed as a visual front end of an AVASR system in vehicle environments. First, we address one essential first step – accurately and reliably locate the face in a moving car. In this work, both color and spatial information are exploited to find a face in a given image. A novel Bhattacharyya-based face detection algorithm is used to compare candidate regions of interest with a unique illumination-dependent face model probability distribution function approximation. In the subsequent step, a lip-specific Gabor filter based feature space is then utilized to extract facial features and locate lips within the frame. In both modules, extensive training and test sets from the AVICAR database will be used to justify design decisions and performance.

2. Face detection

Accurate face detection plays a critical role in successful lip localization and subsequent interpretation of the spoken words through extractred lip parameters. The relatively small size and constantly changing shape of lips does not realistically allow for feasible direct lip detection. Coupled with the difficulties introduced by an unconstrained operational environment, a robust, computationally efficient face detection algorithm is desirable to precede lip localization itself. Many facial recognition methods exist, such as the popular face detector proposed by Viola and Jones in 2001 [10]. However this and many other detectors requires only the intensity component of an image without taking full advantage of the inherent color information which is readily available in most images. In addition, they tend to break down in imagery with complex background such as the database in this study. We believe color could be used as a far more efficient criteria that could drastically reduce the search area and simplify the face detection process. The following sections offer a fast and noise-resistant face detection algorithm by which skin is first classified in an appropriate color space and then subsequently classified as a face or non-face.

2.1 Skin classification via sHSV color space

To determine the optimal color space for efficient skin and face detection, various color spaces have been examined, such as RGB, nrgb, YcbCr, YIQ, and HSV in [11]. Manaully drawn lip masks were constructed from a database of over 400 images that were subsequently used to develop statistical models of Lip, Non-lip, and Skin classes. Histograms were generated for each class and color space and, when applicable, the Gaussian approximations are calculated. Fig.1 shows the approximated Gaussian distributions for each of the three components in five color spaces. Each color spaces'



(a)

(b)

(c)

(d)

(e)

Fig. 1. Gaussian distributions for (a) RGB, (b) NRGB, (c) YCbCr, (d) YIQ, and (e) sHSI color spaces.

components are then compared among the three regions to determine the minimum correlation between non-lip, skin and lip regions. Referencing Fig. 1, while the low variances of the (r,g,b), (Cb, Cr), and (I,Q) components provide a relatively uniform representation for the given region, the non-lip, skin, and lip regions are highly correlated (demonstrated by the overlap seen in the Guassian distributions). Therefore, these components are poor classifiers for discerning lips and skin from that of the background. Additionally, (R,G,B), (Y_{cb}) and (Y_{IQ}) show high correlation between the regions, resulting in a similarly poor classifier. The hue component, on the other hand, provides the maximum separation between skin and non-skin regions and, therefore, is the strongest classifier. Since face images in the database were taken under varying illumination conditions and for different skin tones, hue also provides an illumination-independent and race-independent component, making it ideal for simple, uniform-color thresholding for skin classification. Because of the hue's color wheel effect, to simplify thresholding operations, the standard hue is shifted to the right by a value of 0.2 (72°), resulting in a shifted HSV color space, or sHSV, where region of interest (skin color) incurs no discontinuity. By deploying Bayesian classifier, optimal decision boundaries for the classification can be determined [12]. Fig. 2 illustrates the un-normalized posterior hue distributions, where shifted hue for skin class is approximated by N (0.34, 0.11^2) and the non-skin class by N (0.55, 0.17^2). Here the green lines represent the zero-dimensional decision boundaries that separate the skin and non-skin regions. Between these boundaries, from a shifted hue value of 0.052 to 0.325, the skin posterior distribution surpasses that of non-skin and will classify as a skin pixel.

Building upon the theoretical Bayes classifier, the final skin classification system adds robustness and decrease computational requirements for subsequent face detection. To promote skin region continuity, a hysteresis threshold that uses both spatial and hue information was then employed. Additionally, to increase the skin detection robustness in low-light conditions, a minimum value component of 0.2 is set for all skin pixels, due to the study showing that more than 90% of skin exists above luminosities value of 0.15 when approximated by a Gaussian distribution [12]. To decrease computational demands, the original input image is downsampled to reduce computational complexity when these operations are performed.

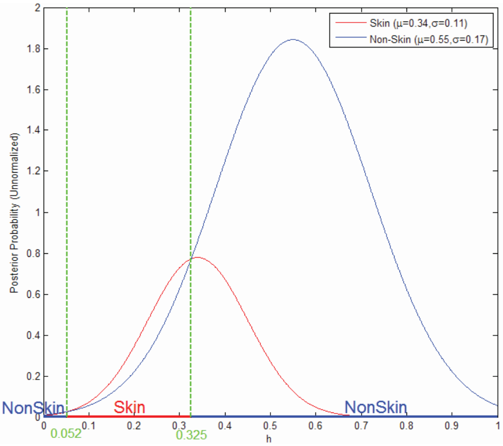


Fig. 2. Un-Normalized Posterior Distributions for Skin and Non-Skin Classes

2.2 Filtering and binary clustering

The unprocessed skin-classified binary images suffer from two main undesirable effects. One type of the error includes single-element impulse noise existing throughout the binary image as false positives within background regions as well as false negatives within skin regions, shown in green boxes in Fig. 3(b). Since false positives were deemed more detrimental to locating the dominant facial skin region, a 33rd percentile order-statistic filter of size 3x3 was selected as a more appropriate filter than the 50th percentile standard median filter. An extra benefit of this filter is that it better separates facial skin regions with skin colored car backgrounds. The red bounding box in Fig. 3(b) illustrates such a boundary, which is preserved via the 33rd percentile filter from (b) to (c). Had a median filter been applied to this image, the segregation would have disappeared and complicated face candidate localization and subsequent face detection. This is an important performance increase as the cluttered and similarly colored car backgrounds often result in false skin detection.

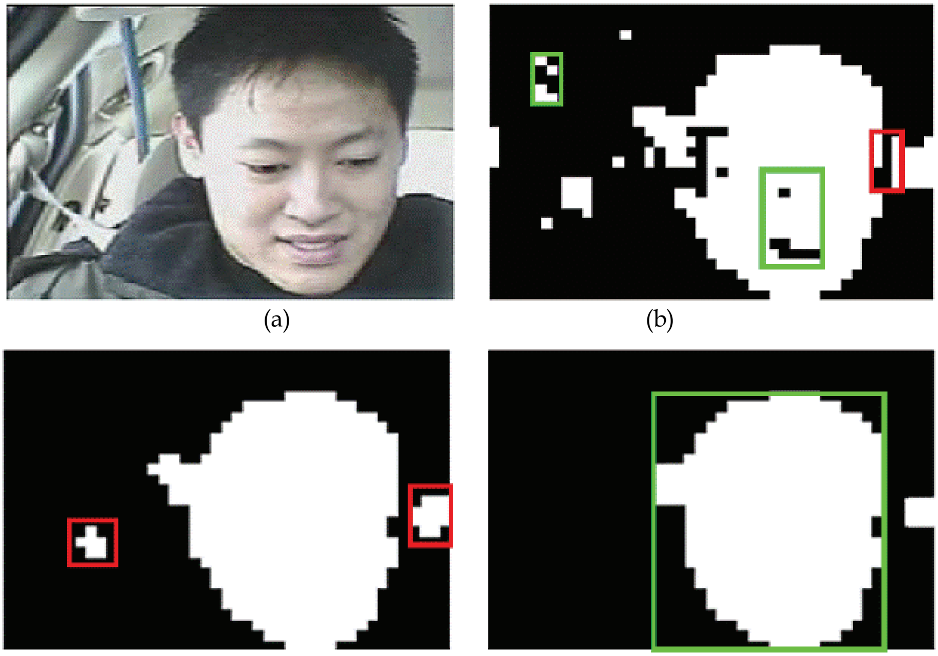

(a)
(b)
(c)
(d)

Fig. 3. Sample Post-Processing Imagery by Step (a) Original Image (b) Skin Classified Binary Image (c) 33rd Percentile Filtered (d) Application of Opening Operation.

The second type of error includes larger, false-positive regions that tend to dominate background (non-skin) regions. The binary morphological operation opening is utilized to minimize this effect. Notice the elimination of the leftmost background cluster in (c) and the reduction in size of the rightmost cluster. Since one face is assumed in each image, the largest skin cluster is selected as the region of interest, shown as the green bounding box in (d), via the connected component labeling. This cluster will now be the input to the subsequent face localization algorithm.

2.3 Face candidate localization algorithm

Despite the filtering and classification methods employed, large regions of falsely classified background pixels still comprise part of the largest cluster returned by the pre-processing algorithm outlined in Section 2.2. Resulting from the unconstrained environment, these problem regions include skin-colored car interior regions, such as a car's roof, and window areas. Fig. 4 illustrates one such distinct, false positive protrusion resulting from a skin colored brick wall behind the car's back windshield. The goal of the face candidate localization algorithm is to simply determine such falsely classified regions attached to the largest cluster and remove them from the region of interest's (ROI's) bounding box. Fig. 5 provides an face candidate localization algorithm flow diagram to be developed within this section.

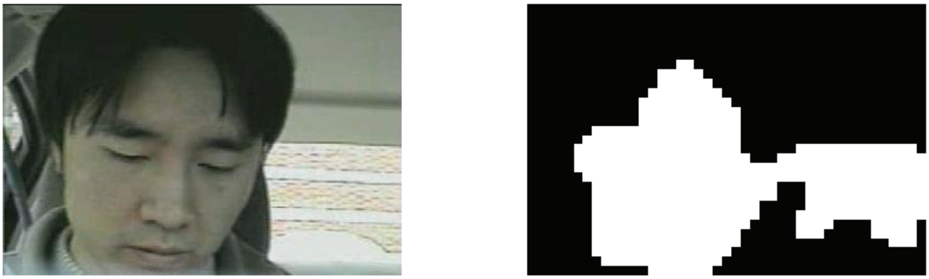


Fig. 4. Example Face Candidate Protrusion

Per Fig.5, the M_cxN_c binary image face candidate, BW_c, is first input to the algorithm. To more effectively separate face candidates without significant background inclusions, an initial candidate screening takes place at the beginning of the algorithm. Sources cite that the average height-to-width ratio of the human face is approximated by the well-known golden ratio of 1.618:1. Accounting for facial tilt and out-of-frame rotation, typical face candidate ROI height-width ratio were found to exist between values of 1.2:1 and 1.7:1 through database measurements over the test subset. Hence, all face candidate ROI's whose height-to-width ratio, M_c/N_c, does not fall within the range [1.2, 1.7] will be subject to the remainder of the ROI pruning process.

For images which fall outside of the acceptable height-width ratio, further filtering takes place. To eliminate clear protrusions which are comparable in size to the face region itself a two-pass spatial filtering technique was employed. This technique locates sudden deviations in cluster configuration between the top and bottom of the face candidate cluster. While other more accurate methods, such as flood-fill techniques, exist to segment binary clusters, these methods are more computationally intensive, requiring several iterations of initial condition- and parameter-dependent morphological operations. Hence, the following computationally inexpensive method was employed to roughly locate distinct binary cluster protrusions similar to that in Fig.4, while preserving the roughly vertically oriented elliptical face region.

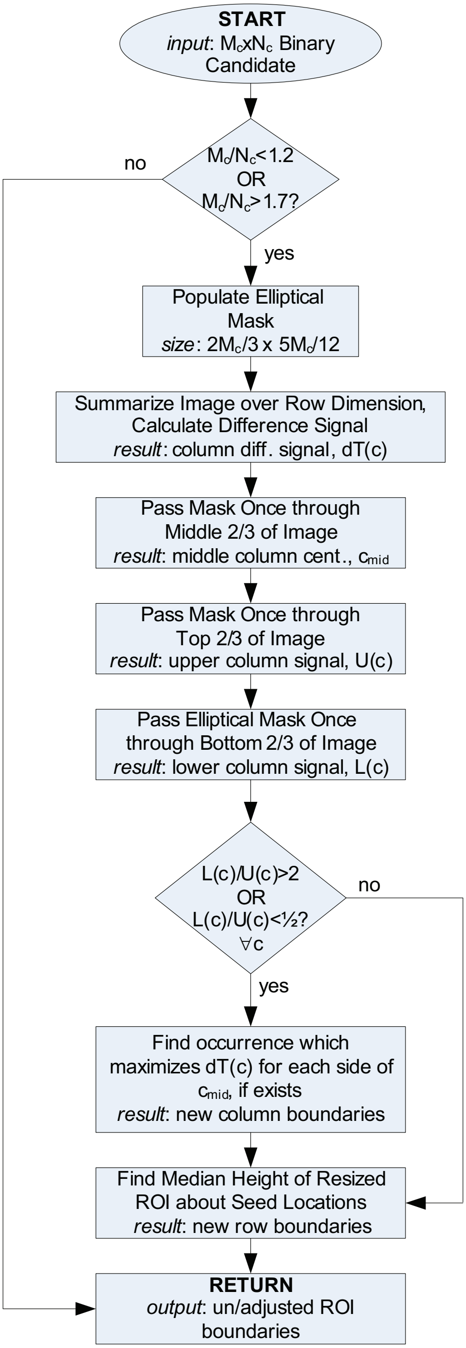

START
input: McxNc Binary Candidate
Mc/Nc<1.2 OR Mc/Nc>1.7?
no
yes
Populate Elliptical Mask
size: 2Mc/3 x 5Mc/12
Summarize Image over Row Dimension, Calculate Difference Signal
result: column diff. signal, dT(c)
Pass Mask Once through Middle 2/3 of Image
result: middle column cent., cmid
Pass Mask Once through Top 2/3 of Image
result: upper column signal, U(c)
Pass Elliptical Mask Once through Bottom 2/3 of Image
result: lower column signal, L(c)
L(c)/U(c)>2 OR L(c)/U(c)<½? ∀c
no
yes
Find occurrence which maximizes dT(c) for each side of cmid, if exists
result: new column boundaries
Find Median Height of Resized ROI about Seed Locations
result: new row boundaries
RETURN
output: un/adjusted ROI boundaries

Fig. 5. Face Candidate Localization Algorithm Flow Diagram

The spatial filtering discussed is the result from passing an elliptical binary mask once through the top two-thirds and bottom two-thirds of the face candidate binary image, BW_c. The height of the elliptical binary mask, called H, was chosen to be two-thirds the input candidate ROI's height, M_c. The width of the ellipse was chosen to mimic the average dimensions of the human face, which is 1.6 times less than its height. Hence, the final size of the elliptical mask is M_h x N_h, where $M_h = floor(2M_c/3)$ and $N_h = floor(5M_c/12)$. The composition of the mask, H, is defined per the following equation

$$H(\mathbf{z}) = \begin{cases} 1 & \text{if} \quad \mathbf{z} \cdot \mathbf{z}^{\mathbf{T}} < 1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\mathbf{z} = \left[\dfrac{r - r_{H,cen}}{r_{H,cen}} \quad \dfrac{c - c_{H,cen}}{c_{H,cen}} \right]$, $c_{H,cen} = N_h / 2$, and $r_{H,cen} = M_h / 2$

where r and c are the row and column location of the elliptical mask. Thusly defined, the elliptical mask is not convolved with the face candidate binary image in the strictest sense. Rather, the elliptical mask, H, is passed once through the top two-thirds and once through the bottom two-thirds of the candidate ROI, centered about one-thirds and two-thirds of the candidate ROI's height, respectively. At each column location, the mask and image are multiplied and then summed by element, returning a value equivalent to the total number of skin-classified pixels enclosed within the mask H at that location. Let $U(c)$ and $L(c)$ be the column signals resulting from the upper and lower passes through the candidate ROI, BW_c, respectively. To preserve the accuracy of the spatial filtering, it should be noted that the input binary image, BW_c, was padded column-wise with $N_h/2$ zeros on each side of the largest cluster. Then the ratio of the upper signal to the lower signal is given by:

$$R(c) = \frac{U(c) + \varepsilon}{L(c) + \varepsilon} \qquad c = 1, 2, \ldots, N_c \tag{2}$$

where ε is a small positive integer introduced to safeguard against $L(c)=0$. This ratio signal effectively shows the relative distribution of the face candidate cluster with $R(c)>1$ indicating a greater concentration at the cluster's top and with $R(c)<1$ indicating a greater concentration at the cluster's bottom. Fig. 6 (a) contains an annotated example of the relative size and shape of the elliptical mask, the resulting upper and lower column signals, U(c) and L(c), as well as the ratio signal, R(c). Note that for clarity this example normalizes each column signal to the area of the elliptical mask.

After the ratio signal has been calculated over the width of the binary image, the binary image is summed across the row dimension yielding a total column vector, T(c). Equivalently, this total signal can be expressed as

$$T(c) = \sum_{r=1}^{M_c} BW_c(r,c) \qquad c = 1, 2, \ldots, N_c \tag{3}$$

Where r and c are the row and column indices, respectively, from the face candidate binary image. Next, an absolute difference signal, $dT(c)$, is derived from T(c) per the following equation:

$$dT(c) = abs\left(T(c+1) - T(c)\right) \qquad c = 1,2,\ldots,N_c - 1 \tag{4}$$

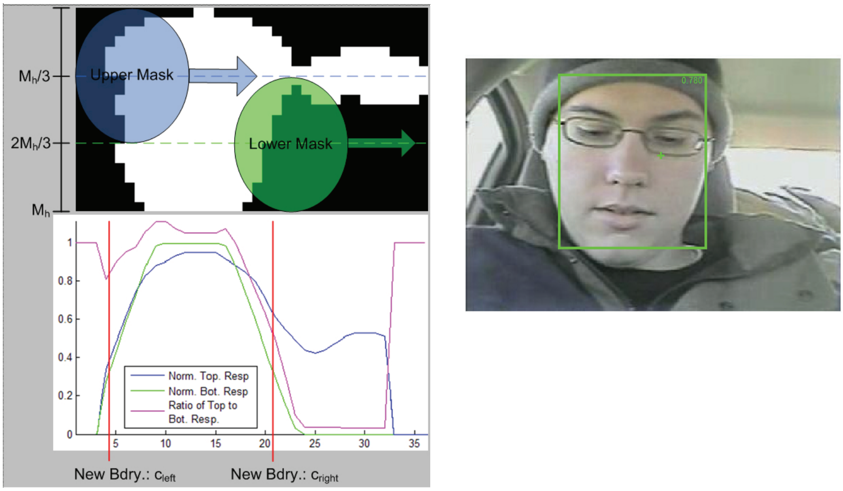
Next, a value of two is chosen to select the factor by which the upper and lower signals can deviate and still be considered part of the facial region. Then, letting **C** be the set of all column locations for which $R(c)$>2 or $R(c)$<0.5, the new horizontal boundaries, $c_{c,left}$ and $c_{c,right}$, of the candidate ROI is then selected by the following equation.

$$c_{c,left} = \begin{cases} \arg_c \max\{dT(c) \mid c \in \mathbf{C}\} \quad 1 \le c \le c_{mid} & \text{if} \quad \mathbf{C} < c_{mid} \ne \varnothing \\ 1 & \text{otherwise} \end{cases}$$

$$c_{c,right} = \begin{cases} \arg_c \max\{dT(c) \mid c \in \mathbf{C}\} \quad c_{mid} < c \le N_c & \text{if} \quad \mathbf{C} > c_{mid} \ne \varnothing \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

where $$c_{mid} = \text{median}\left\{c \mid T(c) = \max\left(T(c)\right)\right\} \qquad c = 1,2,\ldots,N_c$$

where $c_{c,left}$ and $c_{c,right}$ are the new left and right ROI boundaries, respectively, and c_{mid} is the median value of c for which $T(c)$ is maximum over the candidate's entire width. In words, the new boundaries are selected by maximizing the difference signal for all locations where the upper and lower mask differ by a factor of two. This method effectively selects new boundaries located where an abrupt change in top-bottom concentration occurs.



(a) (b)

Fig. 6. Sample Face Candidate Localization Process (a) Original Face Candidate Cluster and Spatial Filter and Ratio Responses; (b) Successfully Modified Bounding Box

Lastly, new top and bottom boundaries, $r_{c,top}$ and $r_{c,bot}$, are created by median filtering the top and bottom cluster edges within $N_c^{'}/20$ pixels of the new ROI's horizontal center. Hence, the new face candidate ROI is now bounded horizontally over [c_{left}, c_{right}] and vertically over [r_{top}, r_{bot}], noting that these ranges are referenced to the origin of the original candidate binary image, BW_c. Fig. 6 (b) illustrates a successfully modified ROI bounding box resultant from this algorithm. Note the correspondence between where the ratio signal drops below one-half and where the new boundaries are located. Also note that these new coordinates are referenced to the downsampled (M_dxN_d) image space and will require conversion back to the original resolution space. Now that the face candidate is localized by its four boundaries, it is subject to the next step, where the face detection algorithm determines whether it indeed is a face.

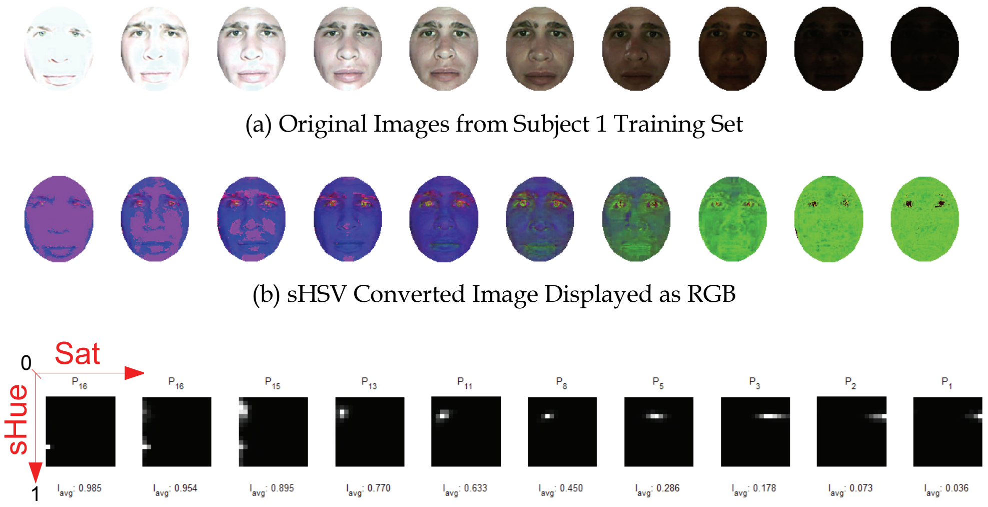
2.4 Face model joint histogram estimation

A critical component of face detection is modeling the variable human face such that a given algorithm provides accurate, repeatable, and reliable results. For this reason, selection of a proper feature set and development of an extensive, representative training set is critical for successful face detection algorithm. Building upon previous work [12], a joint shifted hue and saturation feature space was selected as the basis for face representation since it captures skin color information as well as the variation in saturation incurred around facial features such as eyes, nose, and mouth. Next, the joint probability density function was approximated as a histogram which quantizes the discussed two-dimensional feature space into a finite number of bins. To incorporate spatial information as well, the Epanechnikov kernel is employed in the histogram estimation. The Epanechnikov kernel weights a given ROI heavier towards the center and radially less towards the ROI's perimeter. Hence, it minimizes the effect of background pixels and skin edge pixels which are not always representative of the face itself. Crow utilized the Epanechnikov kernel noting similar advantages and associated performance increases [12]. Another benefit of the Epanechnikov kernel is that it is elliptically symmetric about the ROI's central coordinate, mirroring the natural shape of the human face within the ROI.

2.4.1 Forming the face model joint density estimators

It is observed that while illumination content remains relatively constant within any given image, the average illumination within a given ROI directly impacts the distribution of the face within the joint shifted hue and saturation feature space. Hence, average intensity was chosen as an easily calculable metric which represents the face's ambient lighting conditions. For the sake of consistency, the illumination space was also quantized into a discrete number of bins and the Epanechnikov kernel will weight a pixel's contribution to the average illumination. Borrowing from previous work, the histogram bin count for each feature component, h and s, and the intensity information, I_{avg}, will be segmented into 16 discrete bins uniformly spread about the respective spaces. This value minimizes storage requirements while mitigating the risk of overfitting the actual distribution.

To construct the face model joint density estimators, training set containing 150 images from five individuals of varying skin tone taken under a range of ambient lighting conditions were collected. Care was taken to ensure that across each subject average illumination levels remained within 1/30 of each of the 30 values uniformly spread over the range [0,1]. For each image within the training set, the kernel-weighted intensity and the joint PDF histogram were calculated for each image after conversion to the sHSV color space. Selected results obtained by three of the five subjects are detailed in Fig. 7-9 representing light-, medium-, and dark-skinned individuals, respectively. It can be seen that changes in average



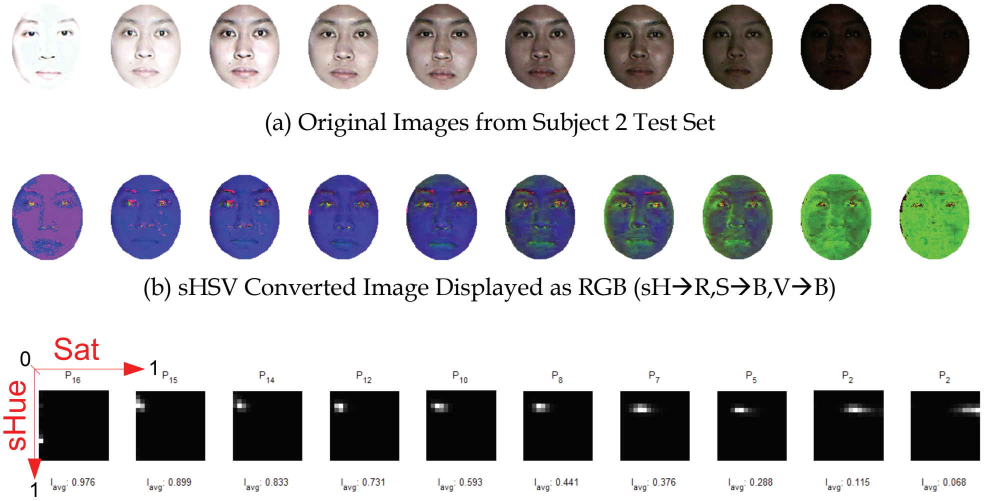
(a) Original Images from Subject 1 Training Set

(b) sHSV Converted Image Displayed as RGB

(c) Corresponding Joint sHue and Saturation Histogram*

**for clarity each histogram has been normalized to its own maximum value*

Fig. 7. Face Model Illumination Dependence Training Set, Subject 1



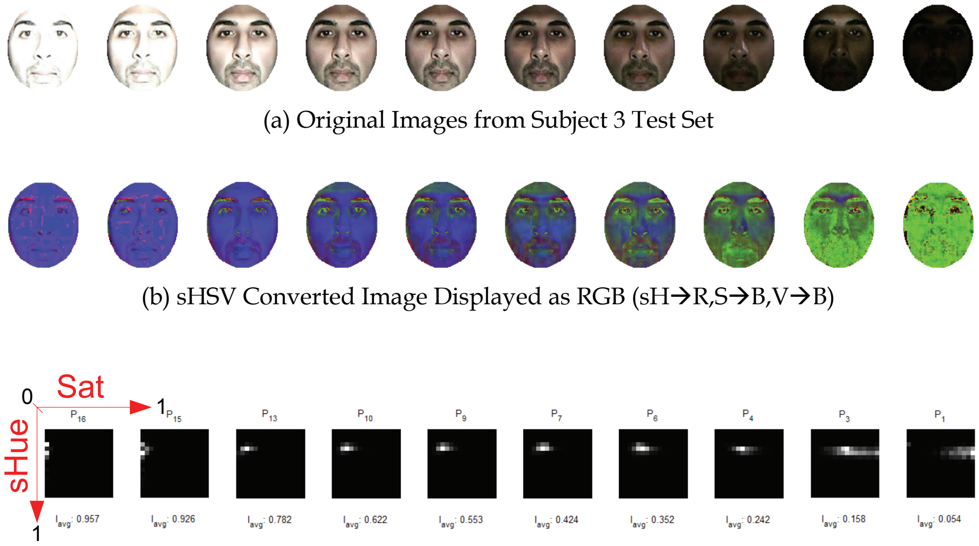
(a) Original Images from Subject 2 Test Set

(b) sHSV Converted Image Displayed as RGB (sH→R,S→B,V→B)

(c) Corresponding Joint Shifted Hue and Saturation Histogram*

**for clarity each histogram has been normalized to its own maximum value*

Fig. 8. Face Model Illumination Dependence Training Set, Subject 2



(a) Original Images from Subject 3 Test Set

(b) sHSV Converted Image Displayed as RGB (sH→R,S→B,V→B)

(c) Corresponding Joint Shifted Hue and Saturation Histogram*

**for clarity each histogram has been normalized to its own maximum value*

Fig. 9. Face Model Illumination Dependence Training Set, Subject 3

illumination directly impact the distribution of the largely unimodal (singly peaked) shifted hue and saturation joint PDF. Furthermore, it can be seen across all PDF histograms that a majority of the hue content is contained within three or four histogram bins across all illumination values. However, saturation content varies from more tightly concentrated at low values under high illumination to roughly three times more spread about the saturation axis under low illumination. Differences in the PDF histograms between light and dark skin tones were slight, involving a positive one-bin shift of the general unimodal distribution along the hue axis. Moreover, at high illumination levels spreading about the hue axis occurred largely due to overexposure at the imaging device itself. Hence, the decision was made to replicate this dependence in the final face model.

The entire 150-image training database was utilized to construct a joint shifted hue and saturation histogram-estimated PDF for each discrete ROI average illumination bin. In words, the face model hisgram set is derived by summing each histogram over the training set whose parent image has the average illumination level and then normalizing each illumination level's PDF histogram independently to unity. The resulting face model PDF histogram approximation across each illumination level is displayed in Fig. 10. Here the value of I_{bin} refers to the average illumination component value which corresponds to the center (midpoint) of the discrete illumination bin, i. This face model histogram set will be stored in memory to be accessed by the face detection algorithm discussed in Section 2.5 to follow.

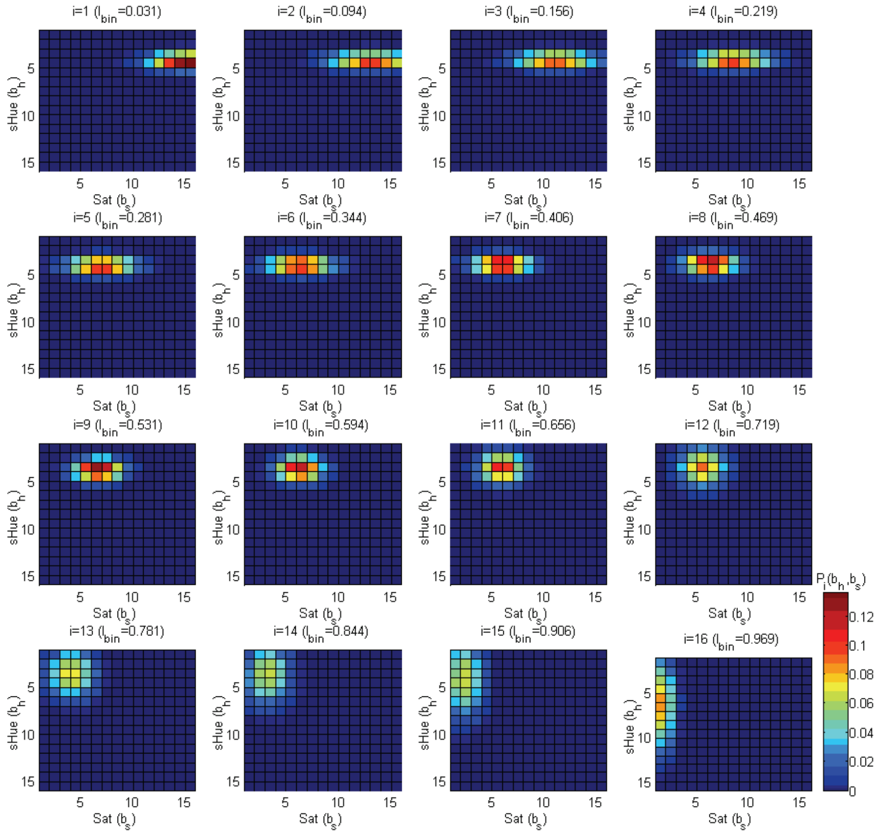


Fig. 10. Joint sHue and Saturation Histogram-Estimated PDF's over Average Illumination Bin Number

2.4.2 Forming the face candidate joint density estimators

With the face model density estimate in place, the face candidate density joint PDF must be constructed so that it can be compared with the model distribution. Derivation of the candidate's histogram approximated joint PDF is straightforward as it only entails the histogram associated with one ROI and its corresponding average illumination value. To complete this task, the face candidate which results from the face candidate localization algorithm (see Section 2.3) is converted to the original coordinate and resolution space. Next, the converted sHSV ROI will be kernel weighted and the histogram estimation process will take place. This face candidate joint density estimate, $\mathbf{P}_i$, will be compared with the face model histogram of the same illumination level, $\mathbf{Q}_i$, via the face detection algorithm outlined in the next section.

2.5 Face detection and test results

With a face model and candidate distributions in hand, candidate ROI's output from the skin detection and filtering algorithm can now be processed for the presence of a face. The

face detection algorithm implemented in this work utilizes the Bhattacharyya coefficient as a means by which the similarity between the generated face model joint histogram and that of a candidate ROI is measured.
The major advantage of the Bhattacharyya coefficient is that, unlike the Mahalanobis distance, it requires no statistical measure from each distribution, drastically reducing computational time and complexity. Remapping the definition of the Bhattacharyya to two dimensions, the Bhattacharyya coefficient can be defined as

$$\rho(\mathbf{P},\mathbf{Q}) = \sum_{h=1}^{m}\sum_{s=1}^{n}\sqrt{P(h,s)\cdot Q(h,s)} \tag{6}$$

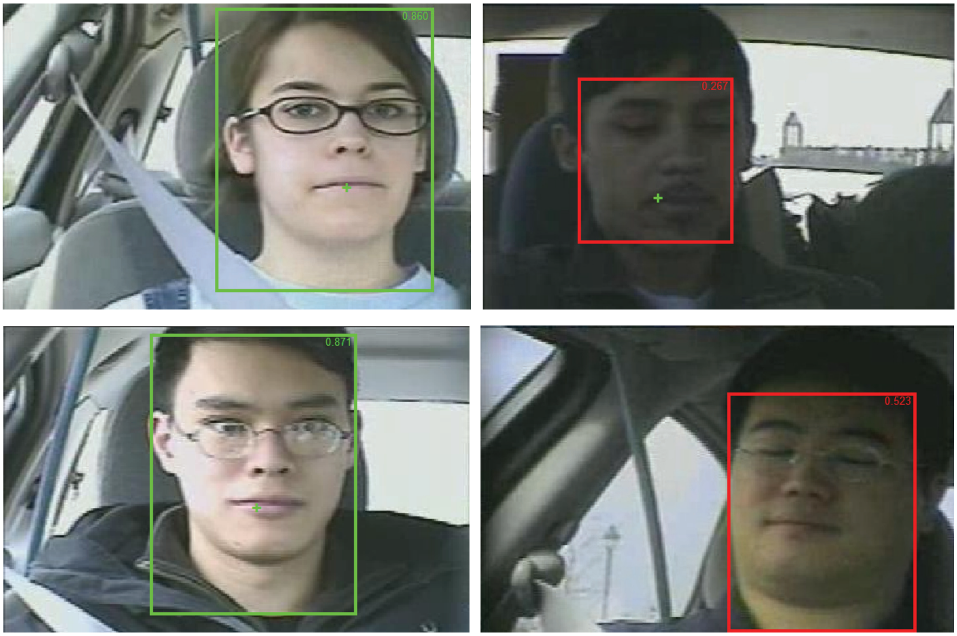
where $\rho(\mathbf{P}, \mathbf{Q})$ is the Bhattacharyya coefficient between the m-by-n bin candidate histogram **P** and m-by-n bin model histogram **Q**, and $P\ (h,s)$ and $Q\ (h,s)$ are the density of the candidate and model histograms, respectively, at bin location $[h, s]$. After the Bhattacharyya coefficient for a given set of candidate and model histograms has been calculated, a simple threshold is applied in order to claffiy the candidate ROI as either a *Face* or a *NonFace*. As expected, false positive error rates decreases as the threhold was increased as higher threholds effectively increased he similarity measure relative to the face model required for face detection. Conversely, false negative failure rates increased as the threshold was increased as an increased number of candidates failed to adequately compare in similarity to the model distribution. Via iterative analysis over the training set composed of 160 images, the Bhattacharyya coefficient threshold of 0.5 was then selected to minimize false negative and false positive error rates.

Face Detection Result	Successful Localization Set*		Complete Test Set	
	Instances	Percentage	Instances	Percentage
Positive Face Detection (ρ ≥ 0.5)	139	94.6%	144	90.0%
Negative Face Detection (ρ < 0.5)	8	5.45%	16	10.0%
Total Images	*147*		*160*	

**successful localization is defined as ROI contains 75% to 125% of the visible face.*

Table 1. Face Detection Algorithm Results

To test the performance of the face detection algorithm, another 160-image test set was created from the AVICAR database, not containing any images found in the face model or skin classification training sets. The test set was composed of 40 subjects at four different time instances throughout the video data. The performance of the face detector using this test set illustrates the success of the algorithm in response to variation in the subject's skin tone as well as any lighting or background changes over time. Recall that this test set also generated the face localization results from Section 2.3, where 147 of the 160 images incurred successful face localization. Table 1 details the true positive and false negative detection rates for both the complete test set and the subset for which the face candidate was successfully localized. As seen, the face detection algorithm achieved an overall accuracy of 90% across all test set images. The accuracy of the algorithm improves by 5% when the face itself is successfully bounded as a result of the face localization algorithm. Sample positive (*Face*) and negative (*NonFace*) classifications are contained within Fig.11 (a) and (b), respectively.



(a) (b)

Fig. 11. Sample (a) Positive Face and (b) Negative Face Detections

3. Lip feature extraction

The Gabor filter is a linear filter whose impulse response is defined as a sinusoidal function multiplied by a Gaussian function in the following form

$$G(x,y\,|\,\theta,F_o,N_x,N_y,\gamma,\eta,\phi)=\frac{\gamma\cdot\eta}{\pi}e^{-\left((\alpha x_r)^2+(\beta y_r)^2\right)}e^{j2\pi F_o\left(x_c\cos\theta+y_c\sin\theta+\phi\right)}$$

$$\forall x\in[1,N_x],\ y\in[1,N_y]$$

with $$\alpha={}^{F_o}\!/_{\gamma},\ \beta={}^{F_o}\!/_{\eta},\ x_o={}^{N_x}\!/_{2},\ y_o={}^{N_y}\!/_{2} \tag{7}$$

where N_x and N_y are the width and height of the Gabor filter mask, respectively, ϕ is the phase of the sinusoid carrier, F_o is the digital frequency of the sinusoid, θ is the sinusoid rotation angle, γ is the Along-Wave Gaussian envelope normalized scale factor, and η is the Wave-Orthogonal Gaussian envelope normalized scale factor. These parameters define the size, shape, frequency, and orientation of the filter among other characteristics. G is the N_y-by-N_x Gabor filter and $[y,x]$ is the spatial location within the filter. The Gabor filter's invariance to illumination, rotation, scale, and translation, and its effective representation of natural images, make the filter an ideal candidate for detecting the facial features in less than desirable circumstances [13].

Utilizing the 160-image training set from the AVICAR database, measurements of upper and lower lip thicknesses and orientations were recorded. It was found the upper lip thickness ratio h_{hi}/ M_c and lower lip thickness ratio h_{low}/ M_c, yield an average value of 0.136 and 0.065, respectively, where M_c measures height of the candidate's facial bounding box. Lip orientation, $\Delta\theta_{lip}$, was recorded as the absolute rotation of the mouth opening axis from horizontal and has an average measurement of 11.25^0. With this data, the Gabor filter set can now be created to more accurately represent the lip region. The final 12-component Gabor filter set, **G**, is thus defined as,

$$\mathbf{G} = \left\{ G_{n,t,f} = G(x,y \,|\, \theta = \theta_t, F_o = F_f, N_x = N_n, N_y = N_n, \gamma, \eta, \phi) \right\}$$

$$\begin{aligned} & N_n \in \left\{ \text{floor}\left(\frac{M_c}{8}\right), \ \text{floor}\left(\frac{M_c}{4}\right) \right\} \quad n = 1,2 \\ & \theta_t \in \left\{ \frac{3\pi}{8}, \ \frac{\pi}{2}, \ \frac{5\pi}{8} \right\} \quad t = 1,2,3 \\ & F_f \in \left\{ \frac{4}{N_n}, \frac{8}{N_n} \right\} \quad f = 1,2 \end{aligned} \tag{8}$$

with $\gamma = \eta = 1$ and $\phi = 0$

where G is defined in Eq. (8) and n, t, and f are the set indices of the (square) Gabor filter size, sinusoid angle, and digital frequency sets, respectively. In words, the Gabor filter set, **G**, is the set of Gabor filters for every combination of n, t, and f. The orientation values, $\theta_{t \in 1,2,3}$, were chosen such that the sinusoid orientation was vertically oriented (θ=90°) and $\pm 2\Delta\theta_{lip}$ away from vertical, where the factor of two was experimentally determined. In addition, the Gabor filter's size, N_n-by-$N_n \,|_{n \in 1,2}$, was selected such that over 80% of the total energy contained in the unbounded Gabor filter is contained within the N_n-by-N_n mask for any value of F_f (which depends upon N_n) and θ_t. The relative size and frequency of the Gabor filter to the candidate's height allows for a more scale-invariant design.

3.1 Gabor filtering algorithm

With the establishment of the lip-specific Gabor filter set, processing of the face-classified region of interest can proceed. Here, the sHSV triplet's value (illumination) component is selected as the feature space of choice for Gabor filtering since it best separates lip and surrounding face.

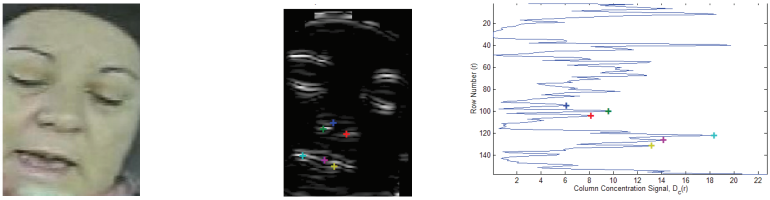
First, 12 Gabor filter responses are generated by performing two-dimensional convolution of the face-classified image's value component, V, independently with each Gabor filter configuration. Next, all 12 Gabor responses are summarized element by element. Due to the positive- and negative-valued modes of the Gabor filters, the total Gabor response is then normalized to the range [0,1] and further remapped to stress the maximal and minimal Gabor jet values. The final, normalized, and remapped Gabor filter response is denoted as

G_f, and has size M_c-by-N_c where M_c and N_c are the row and column sizes of the face candidate, respectively.

Subsequently, the Gabor filter response, G_f, undergoes mean-removal where all response pixel values are set to zero if they are less than the total response's sample mean and are left unchanged if the values are above the mean. Furthermore, to remove false positives within the background surrounding the face, the skin-classified binary mask is applied over the mean removed response. Fig. 12(b) shows a mean-removed and masked Gabor response, $G_{mr,}$ of the original image in (a). As can be seen, smooth skin surfaces, such as the cheeks, provide minimal response while the mouth opening, lips, nostrils, eyes, and eyebrows provide much elevated responses. In addition, the cross section of the lip from chin to the region above the lip involves many oscillatory changes in intensity value. Mean removal effectively eliminates the contribution of background pixels to subsequent processing. The skin-classification masking also noticeably reduces the effect of several high-intensity non-face background regions.

3.2 Lip center coordinate estimation

Given the mean-removed and masked Gabor response, G_{mr}, a number of possible lip locations, called seeds, will be generated. Here, a column concentration signal, D_c, is first calculated from the G_{mr}. Then, seed row coordinates, $r_{pk,i}$ are chosen as local maxima of D_c, see colored crosses in Fig. 12(c). Peaks above image mean row value which do not exceed signal's mean are discarded. Finally, seed column coordinates $c_{pk,i,}$ are chosen as midpoint of longest nonzero response chain in row. Hence, the i^{th} seed point now has the location $[r_{pk,i},c_{pk,i}]$. Fig. 12(b) shows the Gabor response, G_{mr}, overlaid with the seed locations indicated by the colored crosses.



(a) (b) (c)

Fig. 12. Sample Lip Coordinate Estimation Process. (a) Original RGB Face Candidate (b) Seed Locations within Mean-Removed, Masked Gabor Response (c) Seed Row Locations Overlaid on Dc Plot

Following seed generation, key parameters which are indicative of the presence of lips will be calculated. Utilizing these parameters, the *figure of merit* (FOM) will then be calculated as

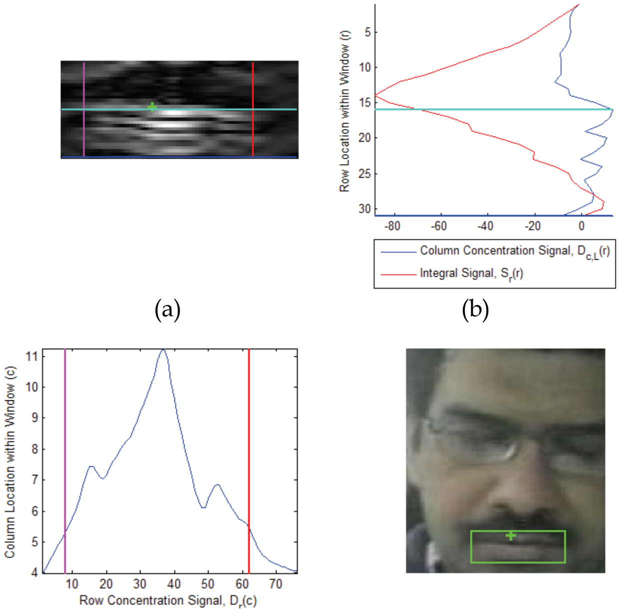
$$\begin{aligned}&\mathbf{FOM} = \{FOM_i\} = \left\{D_{loc,i} \cdot D_{pk,i} \cdot r_{pk,i}\right\}\\&D_{loc,i} \geq 1,\ \ D_{pk,i} \geq 1,\ \ r_{pk,i} \in [\mu_r, M_c]\end{aligned} \tag{9}$$

where **FOM** is the set of all figure of merit values, FOM_i, at seed index i, D_{loc} is the local two-dimensional concentration of G_{mr} about the seed, D_{pk} is the sum of all column concentration signal peaks about the seed, and r_{pk} is the seed row location. Conceptually, the figure of merit in Eq. (9) combines the most visually apparent features of the lips into a single function. It has been argued that the lip's central coordinates are the coordinates for which the established figure of merit is maximal.

The lip center coordinate estimator was applied to the test set used in the previous sections. It was found that the figure of merit and Gabor filter system utilized in the lip coordinate estimate yields comparable results to those of the face detector algorithm of Section 2. Of the 139 images for which the face candidate ROI was successfully localized and classified as a face, the algorithm placed the lip coordinates on the lips for 89.2% of the time. When applied to the test set in its entirety, the lip coordinate estimation algorithm placed the estimated coordinate on the lips 83.8% of the time.

3.3 Lip localization and test results

Vertical lip localization within an image is inherently more complex than horizontal localization due to the striation (layers) of the Gabor response in the lip axis direction. Due to this, horizontal lip localization will be performed first to increase accuracy of the vertical localization. Fig. 13 illustrates lip localization procedure. To locate the lips in the horizontal axis, the row concentration signal $D_r(c)$ is computed over the lip region, shown in (c). Then, the left and right boundaries are determined where $D_r(c)$ is at 10% of that signal's maximum value above the mean.



(a) (b)

(c) (d)

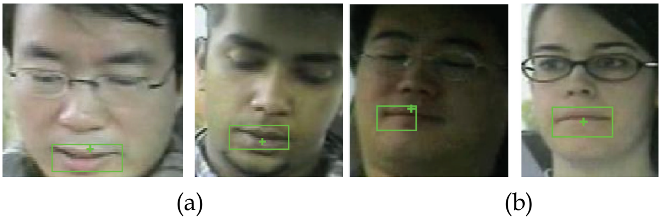
Fig. 13. Sample Horizontal and Vertical Lip Localization Procedure and Result. (a) Gabor Response within Lip Region (b) $D_{c,L}$ and S_r Signals over Lip Region Row (c) D_r Signal over Lip Region Column and (d) Lip Localization Result

After horizontal lip localization, vertical localization is undertaken, utilizing the returned left and right boundaries. To do so, the column concentration signal, $D_{c,L}(r)$, and column discrete integral signal, $S_r(r)$, are calculated between the left and right bounds only (see (b)). The integral signal is the summation of the mean-removed column concentration signal from the top of the lip region to row index r. Mean subtraction was performed on the column concentration signal such that lower intensity regions (rows) of pixels would count negatively toward the integral signal and higher intensity regions would positively count toward the signal. Finally, the lip localized upper and lower boundaries are found where the points are at 10% of S_{max} above the upper and lower minimum values, respectively. Sample lip localization success and failures are shown in Fig.14(a) and (b), respectively. When applied to the 160-image test set, factoring in face detection, the overall accuracy of 75.6%. Note that if the detected lip boundary is more than 5 pixels away from the lip corner or the closest lip point vertically, it is considered as a failure. The last image in Fig. 14 is considered a failure because the detected region contains more than 125% of the actual lips. While the overall accuracy is less than ideal, the challenges of the sub-optimal image quality and the unconstrained car environment make this a respectable value.

4. Conclusion and future work

Relative to previous work, positive face detection rates rose from 75% to 90% while effective lip localization rates rose from 65% to 75% when considering face detection as a front end to lip localization [12]. Among many techniques considered, the unique illumination-dependent face model and the adjusted skin classifier are considered successful and critical to the stated performance increase in face detection. The lip localization algorithm proposed a unique Gabor response feature space which relied upon a figure of merit rather than heuristic approximations, making it more versatile within the unconstrained environment.

Despite the stated performance increases, common sources of error include limited image resolution, skin-colored car environments, and overly bright and dark operating conditions without sufficient image dynamic range. The most notable improvement to the lip localization algorithm would be realized through the inclusion of time into the algorithm. Advanced difference imaging, the detection of movement between frames, would improve face localization and detection while reducing additional processing. Furthermore, face and lip spatial movement are generally orthogonal to each other, aiding the lip localization process even further.



(a) (b)

Fig. 14. Sample Lip Localization (a) Success and (b) Failures

5. References

[1] Stork, D.G. & Hennecke, M.E. (1996). *Speechreading by Humans and Machines*, in NATO ASI Series F, vol. 150, Sprinter Verlag.

[2] Zhang, X.; Broun, C.C.; Mersereau, R.M.; & Clements, M.A. (2002). Automatic Speechreading with applications to human-computer interfaces, *EURASIP Journal Applied Signal Processing, Special Issue on Audio-Visual Speech Processing*, vol. 1, pp.1228-1247.

[3] Potamianos, G. et al. (2004). Audio-Visual Automatic Speech Recognition: An Overview, in *Issues in Visual and Audio- Visual Speech Processing* by G. Bailly, E. Vatikiotis, and Perrier, Eds, MIT Press.

[4] Liew, A. & Wang, S.L. (2009) . *Visual Speech Recognition: Lip Segmentation and Mapping*, Medical Information Science Reference.

[5] Coulon, D.; Delmas, P.; Coulon, P.Y. & Fristot, V. (1999). Automatic Snakes for Robust Lip Boundaries Extraction, in *Proc. ICASSP*.

[6] Luettin, J.; Tracker, N.A. & Beet, S.W. (1995) . Active Shape Models for Visual Speech Feature Extraction, *Electronic System Group Report No95/44*, Univ. Of Sheffield, UK.

[7] Zhang, X. & Mersereau, R.M. (2000). Lip feature extraction towards an automatic speechreading system, *Proc. of IEEE ICIP*

[8] Wang, S.L.; Liew, A.; Lau W.H. & Leung, S.H. (2009). Lip Region Segmentation with Complex Background", in [4].

[9] Lee, B.; Hasegawa-Johnson, M.; Goudeseune, C; Kamdar, S.; Borys, S.; Liu, M.& Huang, T. (2004). AVICAR: Audio-Visual Speech Corpus in a Car Environment, in *INTERSPEECH2004-ICSLP*.

[10] Viola, P. & Jones, M. (2001). Robust Real-time Object Detection, in *Internationa Journal of Computer Vision*.

[11] Zhang, X.; Montoya, H.A. & Crow, B. (2007). Finding Lips in Unconstrained Imagery for Improved Automatic Speech Recognition, in *Proc. 9th International Conference on Visual Information Systems*, Shanghai, China.

[12] Crow, B. & Zhang, X. (2009). Face and Lip Tracking in Unconstrained Imagery for Improved Automatic Speech Recognition, in *Proc. 21th IS&T/SPIE Annual Symposium on Electronic Imaging*, San Jose, California.

[13] J. Kamarainen, V. Kyrki (2006). Invariance Peroperties of Gabor Filter-Based Features—Overview and Applications, in *IEEE Transactions on Image Processing*, vol. 15, no. 5, May 2006, pp. 1088-1099.

14

Visual Speech Recognition

Ahmad B. A. Hassanat
*IT Department, Mu'tah University,
Jordan*

1. Introduction

Lip reading is used to understand or interpret speech without hearing it, a technique especially mastered by people with hearing difficulties. The ability to lip read enables a person with a hearing impairment to communicate with others and to engage in social activities, which otherwise would be difficult. Recent advances in the fields of computer vision, pattern recognition, and signal processing has led to a growing interest in automating this challenging task of lip reading. Indeed, automating the human ability to lip read, a process referred to as visual speech recognition (VSR) (or sometimes speech reading), could open the door for other novel related applications.

VSR has received a great deal of attention in the last decade for its potential use in applications such as human-computer interaction (HCI), audio-visual speech recognition (AVSR), speaker recognition, talking heads, sign language recognition and video surveillance. Its main aim is to recognise spoken word(s) by using only the visual signal that is produced during speech. Hence, VSR deals with the visual domain of speech and involves image processing, artificial intelligence, object detection, pattern recognition, statistical modelling, etc.

There are two different main approaches to the VSR problem, the visemic* approach and the holistic approach, each with its own strengths and weaknesses. The traditional and most common approaches to automatic lip reading are based on visemes. A Viseme is the mouth shapes (or appearances) or sequences of mouth dynamics that are required to generate a phoneme in the visual domain. However, several problems arise while using visemes in visual speech recognition systems such as the low number of visemes (between 10 and 14) compared to phonemes (between 45 and 53). Visemes cover only a small subspace of the mouth motions represented in the visual domain, and many other problems. These problems contribute to the bad performance of the traditional approaches; hence, the visemic approach is something like digitising the signal of the spoken word, and digitising causes a loss of information.

The holistic approach such as the "visual words" (Hassanat, 2009) considers the signature of the whole word rather than only parts of it. This approach can provide a good alternative to the visemic approaches to automatic lip reading. The major problem that faces this approach is that for a complete English language lip reading system, we need to train the whole of the English language words in the dictionary! Or to train (at least) the distinct ones. This

* Related to a Viseme.

approach can be effective if it is trained on a specific domain of words, e.g. numbers, postcodes, cities, etc.

A typical VSR system consists of three major stages: detecting/localizing human faces, lips localization and lip reading. The accuracy of a VSR system is heavily dependent on accurate lip localisation as well as the robustness of the extracted features. The lips and the mouth region of a face reveal most of the relevant visual speech information for a VSR system (see Figure 1).

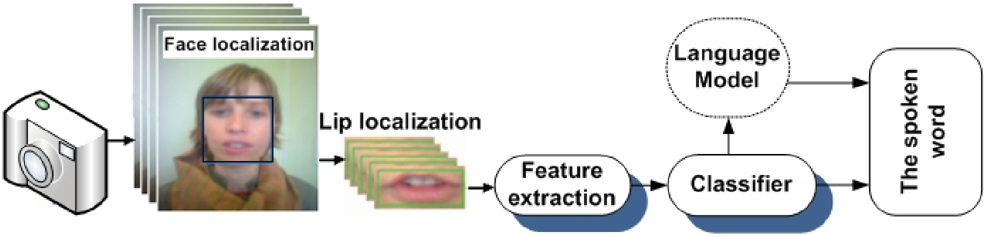


Fig. 1. A typical VSR system.

The last stage is the core of the system in which the visual features are extracted and the words are recognised. Unlike the visemic approach, this study proposes an holistic approach to tackle the VSR problem, where the system recognizes the whole word rather than just parts of it. In the proposed system, a word is represented by a signature that consists of several signals or feature vectors (feature matrix), e.g. height of the mouth, mutual information, etc.

Each signal is constructed by temporal measurements of its associated feature. The mouth height feature, for instance, is measured over the time period of a spoken word. This approach is referred to as the "visual words" (VW) approach. A language model is an optional step that can be used to enhance the performance of the system.

1.1 Human lip reading skills

Lip reading is not a contemporary invention; it was practised as early as 1500 AD, and probably before that time. The first successful lip reading teacher was the Spanish Benedictine monk, Pietro Ponce, who died in 1588. Lip reading teaching subsequently spread to other countries. The German Samuel Heinecke opened the first lip reading school in Leipzig in 1787. The first speech reading conference was held at Chautauqua, USA in 1894 (Bruhn, 1920).

Several different methods have been described in the literature for human lip reading such as the Muller-Walle, Kinzie, and the Jena methods. The Muller-Walle method focuses on the lip movement to produce a syllable as part of words, and the Kinzie method divides lip reading teaching into 3 teaching levels, depending on the difficulty (beginners, intermediate and advanced) (De Land, 1931). Although only 50% or less of speech can be seen, the reader must guesstimate those words that he/she has missed. This was the core of the Jena method: training the eye and exercising the mind (De Land, 1931). However, regardless of the variety of known lip reading methods, all methods still depend on the lip movement that can be seen by the lip reader.

Potamianos et al. (2001) described a human speech perception experiment. A small number of human listeners were presented with the audio once and the audio and video of 50 database sequences from an IBM ViaVoice database single speaker, with different bubble noises added each time. The participants were asked to transcribe what they heard and viewed.

Potamianos et al.'s (2001) experiment is not a pure lip reading experiment, as its aim was to measure the effect of the visual cues on the human speech perception, rather than the perception of the speech without the audio. The experiment showed that human speech perception increases by seeing the video and watching the visual cues. The word error rate was reduced by 20% when participants viewed the video, showing that the human audio-visual speech perception is about 62% word accuracy. According to the previous study, about 30% of the participants were non-native speakers, and this is one of the reasons why the recognition rate was very low, despite both the audio and video signals being revealed.

A human lip reading experiment was conducted in this study to roughly measure the human ability for lip reading, and the amount of information that can be seen from speech. Four video sequences from the PDA Database (Morris, et al., 2006) were used in this experiment, 2 males and 2 females; each video spoke 10 digits; the digits and their sequences are different from one video to another, and the audio signals were removed from the four videos. Fifty five participants were asked to transcript what each video spoke; each participant can play each video up to 3 times, so participants would have enough time to decide what the spoken digits were, and they would not be fooled by the speed of the video. These videos were uttering only digits, {1,2,3,…,9}, the participants were informed about this domain (the speech subject), hence it is much easier for humans to read lips if they know the subject of the talk, and also it mimics automatic lip reading experiments since the recognizer algorithm knows in advance and is trained on all the classes (the words) to be classified; the average word recognition rate for all the participants was 53%. See Table 1.

Subject	Result
1 (Female)	61%
2 (Male)	50%
3 (Male)	37%
4 (Female)	63%
Average	53%

Table 1. Human lip reading results.

As can be seen from Table 1, some videos were easier to read than others (61% and 63% for videos 1 and 4 respectively), where some other videos have less information for lip readers, or those people by nature either speak faster than normal, or do not produce enough information for the lip readers. We can notice that the females give more information for the readers; it is, of course, difficult to substantiate such a claim because this is a small experiment using a small number of videos, so it is too early to draw such conclusions with such evidence. The most important thing that this experiment can reveal so far is the overall human lip reading ability, which is 53%. Another interesting thing to mention is that different people also have different abilities to perceive speech from visual cues only. In this experiment the best lip reader result was 73%, while the worst was 23%. These experiments illustrate the variation in individual lip reading skills, and the variation in individual ability to produce a clear readable visual signal, which would add to the challenge of designing an automatic lip reading system.

The human ability for lip reading varies from one person to another, and depends mainly on guessing to overcome the lack of visual information. Needless to say, lip readers need to have a good command of the spoken language, and in some cases the lip reader improvises and uses his/her knowledge of the language and context to pick the nearest word that he/she feels fits into the speech. Moreover, human lip readers benefit from visual cues detected outside the mouth area (e.g. gestures and facial expressions). The complexity and difficulties of modelling these processes present serious challenges to the task of automatic visual speech recognition.

1.2 In-house video database

Some of the methods discussed in this chapter are evaluated using an in-house video database (Hassanat, 2009). This database consists of 26 participants of different races and nationalities (Africans, Europeans, Asians and Middle Eastern volunteers). Each participant recorded 2 videos (sessions 1 and 2) at different times (about a 2 month period in time between the two recordings). The participants were 10 females and 16 males, distributed over different ethnic groups: 5 Africans, 3 Asians, 8 Europeans, and 10 Middle Eastern participants. Six of the males had both beard and moustache, and 3 males had moustache only.

The videos were recorded inside a normal room, which was lit by a 500-watt light source, using Sony HDR-SR10E high definition (HD) 40GB Hard Disc Drive Handy-cam Digital Camcorder – 4 Mega Pixels. The videos were de-interlaced then compressed using Intel IYUV codec, converted to AVI format, and resized to (320 x 240) pixels, because it is easier to deal with AVI format, and it is faster for training and analyzing the videos with smaller frame sizes. Each person in each recorded video utters non-contiguous 30 different words five times, which are numbers (from 0-9), short look-alike words (*knife, light, kit, night, fight*) and (*fold, sold, hold, bold, cold)*, long words: (*appreciate, university, determine, situation, practical*) and five security related words (*bomb, kill, run, gun, fire*).

1.3 Chapter overview

This chapter consists of 7 sections. In the first section, we presented a brief introduction to the VSR and human ability to read lips, and briefly described a typical VSR system. Section 2 briefly reviews automatic lip reading literature and describes some of the (state-of-the-art) approaches to VSR. Section 3 presents different approaches to face detection/localization. Lip localization approaches are reviewed in section 4. Section 5 is dedicated to the features extraction and recognition method. Some experimental results are presented in section 6. The chapter summary and some conclusions are discussed in section 7.

2. VSR literature review

Most of the work done on VSR came through the development of AVSR systems, as the visual signal completes the audio signal, and therefore enhances the performance of these systems. Little work has been done using the visual only signal. Most of the proposed lip reading solutions consist of two major steps, feature extraction, and Visual speech feature recognition. Existing approaches for feature extraction can be categorised as:

1. **Geometric features-based approaches** - obtain geometric information from the mouth region such as the mouth shape, height, width, and area.
2. **Appearance-based approaches** - these methods consider the pixel values of the mouth region, and they apply to both grey and coloured images. Normally some sort of dimensionality reduction of the region of interest (ROI) (the mouth area) is used such as the principal component analysis (PCA), which was used for the Eigenlips approach, where the first n coefficients of all likely lip configurations represented each Eigenlip.
3. **Image-transformed-based approaches** - these methods extract the visual features by transforming the mouth image to a space of features, using some transform technique, such as the discrete Fourier, discrete wavelet, and discrete cosine transforms (DCT). These transforms are important for dimensionality reduction and to redundant data elimination.
4. **Hybrid approaches**, which exploit features from more than one approach.

2.1 Geometric features-based approaches

A geometric features-based approach includes the first work on VSR done by Petajan in 1984, who designed a lip reading system to aid his speech recognition system. His method was based on using geometric features such as the mouth’s height, width, area and perimeter (Petajan, 1984).

Another recent work in this category is the work done by (Werda et al., 2007), where they proposed an Automatic Lip Feature Extraction prototype (ALiFE), including lip localization, lip tracking, visual feature extraction and speech unit recognition. Their experiments yielded 72.73% accuracy of French vowels, uttered by multiple speakers (female and male) under natural conditions.

2.2 Appearance-based approaches

Eigenlips are the compact representation of mouth Region of Interest using PCA. This approach was inspired by the methods of (Turk & Pentland, 1991), and first proposed by (Bregler & Konig, 1994). Another Eigenlips-based system was investigated by (Arsic & Thiran, 2006), who aimed to exploit the complementarity of audio and visual sources. (Belongie & Weber, 1995) introduced a lip reading method using optical flow and a novel gradient-based filtering technique for the features extraction process of the vertical lip motion and the mouth elongation respectively.

In a more recent study, (Hazen et al., 2004) developed a speaker-independent audio-visual speech recognition (AVSR) system using a segment-based modelling strategy. This AVSR system includes information collected from visual measurements of the speaker's lip region using a novel audio-visual integration mechanism, which they call a segment-constrained Hidden Markov Model (HMM). (Gurban & Thiran, 2005) developed a hybrid SVM-HMM system for audio-visual speech recognition, the lips being manually detected. The pixels of down-sampled images of size 20 x 15 are coupled to get the pixel-to-pixel difference between consecutive frames. (Saenko et al., 2005) proposed a feature-based model for pronunciation variation to visual speech recognition; the model uses dynamic Bayesian network DBN to represent the feature stream.

(Sagheer et al., 2006) introduced an appearance-based lip reading system, employing a novel approach for extracting and classifying visual features termed as "Hyper Column Model" (HCM). (Yau et al., 2006) described a voiceless speech recognition system that employs dynamic visual features to represent the facial movements. The system segments the facial movement from the image sequences using motion history image MHI (a spatio-temporal template). The system uses discrete stationary wavelet transform (SWT) and Zernike moments to extract rotation invariant features from MHI.

2.3 Image-transformed-based approaches

(Lucey & Sridharan's, 2008) work was designed to be posing invariant. Their audio-visual automatic speech recognition was designed to recognize speech regardless of the pose of the head, the method starting with face detection and head pose estimation. They used the pose estimation method described by (Viola & Jones, 2001). The pose estimation process determines the visual feature extraction to be applied either on the front face, the left or the right face profile. The visual feature extraction was based on the DCT, which was reduced by the linear discriminative analysis (LDA), and the feature vectors were classified using HMM.

A very recent study which also fits into this category was done by (Jun & Hua, 2009), where they used DCT for feature extraction from the mouth region, in order to extract the most discriminative feature vectors from the DCT coefficients. The dimensionality was reduced by using LDA. In addition, HMM was employed to recognize the words.

2.4 Hybrid approaches

(Neti et al., 2000) proposed an audio-visual speech recognition system, where visual features obtained from DCT and active appearance model (AAM) were projected onto a 41 dimensional feature space using the LDA. Linear interpolation was used to align visual features to audio features.

A comparative Viseme recognition study by (Leszczynski & Skarbek, 2005) compared 3 classification algorithms for visual mouth appearance (Visemes): 1) DFT + LDA, 2) MESH + LDA, 3) MESH + PCA. They used two feature extraction procedures: one was based on normalized triangle mesh (MESH), and the other was based on the Discrete Fourier Transform (DFT), the classifiers designed by PCA and LDA.

Yu (2008) made VSR the process of recognizing individual words based on a manifold representation instead of the traditional visemes representation. This is done by introducing a generic framework (called Visual Speech Units) to recognise words without resorting to Viseme classification.

The previous approaches can be further classified depending on their recognition and /or classification method. Researchers usually use dynamic time warping (DTW), e.g. the work done by Petajan. Artificial neural networks (ANN), e.g. the work done by Yau et al. and Werda et al.. Dynamic Bayesian Network (DBN), e.g. the work done by Belongie and Weber, and support vector machines (SVM), e.g. the work done by Gurban and Thiran, and Saenko et al.

The most widely used classifier in the VSR literature is the hidden Markov models (HMM). Methods that use HMM include Bregler and Konig; Neti, et al.; Potamianos et al.; Hazen et al.; Leszczynski and Skarbek; Arsic and Thiran; Sagheer, et al.; Lucey and Sridharan; Yu; and Jun and Hua.

Each of the previous approaches has its own strengths and weaknesses. Sometimes the data reduction methods cause the loss of a considerable amount of related data, while using all the available information takes a much longer processing time, and not necessarily to obtain better results due to video or image-dependent information. More effort should be invested to propose any combination of the different approaches, to trade the disadvantages of each individual approach.

Most of the previous studies on VSR contain promising solutions, especially when combining an audio signal with a video signal. Although most of these systems rely on a clean visual signal (Saenko et al., 2005), still, for visual alone speech reading systems or subsystems, they have a high word error rate (WER). Sometimes WER is more than 90% for large vocabulary systems (Hazen et al., 2004; Potamianos, et al., 2003) and a range of 55% to 90% for small vocabulary systems (Yau et al., 2006). The main reason behind this high WER is that VSR problems represent a very difficult task by nature, as the visual side provides little information about speech. Other reasons that increase WER include: Large variations in the way that people speak (Yau et al., 2006), errors in pre-process steps of VSR systems such as face detection and lips localization, visual appearance differences between individuals, particularly, in speaker-independent systems, the visemes problems, and other general problems like light conditions and video quality.

3. Face detection

Face detection is an essential pre-processing step in many face-related applications (e.g. face recognition, lips reading, age, gender, and race recognition). The accuracy rate of these applications depends on the reliability of the face detection step. In addition, face detection is an important research problem for its role as a challenging case of a more general problem, i.e. object detection.

The most common and straightforward example of this problem is the detection of a single face at a known scale and orientation. This is a nontrivial problem, and no method has yet been found that can solve this problem with 100% accuracy. Factors influencing the accuracy of face detection include variation in recording conditions/parameters such as pose, orientation, and lighting. However, there are several algorithms and methods that deal with this problem, attaining various accuracy rates under varied conditions. Most existing schemes are based on somewhat restrictive assumptions. Some of the most successful methods used 20×20 (or so) pixel observation window across the image for all possible locations, scales, and orientations. These methods include the use of support vectors machines (Osuna et al., 1997), neural network (Rowley et al., 1998) or the maximum likelihood approach based on histograms of feature outputs (Schneiderman and Kanade, 2000). Others use a cascaded support vector machine (Romdhani, et. al., 2004). Some researchers use the skin colour to detect the face in coloured images (Garcia, & Tziritas, 1999).

In their study, (Yang et. al., 2002) classified face detection methods in still images into four categories:

1. Knowledge-based methods. These methods require human knowledge about facial features.
2. Feature invariant approaches. Designed to find structural features that are not affected by the general problems as with the face detection process, such as pose and light conditions. The targeted features vary from one researcher to another, but mostly they

concentrated on facial features, texture, skin colour, or a combination of the previous features.

3. Template matching methods. Using one or more patterns to describe a typical face, then comparing this pattern with the image to find the best correlation between the pattern and a window in the targeted image. These templates can be predefined templates or deformable templates.
4. Appearance-based methods. Like the previous approach, but the template is not previously declared, rather it is learned from a set of images, then the learned template is used for detection. A variety of methods fill in this gap, such as: Eigenface (e.g. Eigenvector decomposition and clustering), distribution-based (e.g. Gaussian distribution), see (Sung & Poggio, 1998; Samaria, 1994), Neural networks, support vector machines, Hidden Markov Model, Naïve Bayes classifier, and information-theoretical approach.

Others classified face detection methods into two approaches: features-based approaches and image-based approaches (Hjelmas, & Low, 2001). The problem with most of these methods is that they are very sensitive to variation in light conditions and complex backgrounds. However, the one proposed by Rowley et al. (1998) for face detection is one of the best face detection methods created so far, and is used as a benchmark test by many researchers.

Rowley's et al. (1998) method for face detection consists of two stages, first applying a neural network-based filter that receives a 20 x 20 pixel region of the input image, and output values ranging from -1 to 1, which means non-face or face respectively. Assuming that faces in an image are upright and looking at the camera, the filter is applied to all locations in the image, to obtain all the possible locations of the face. To solve the scale problem, the input image is repeatedly sub-sampled by a factor of 1.2. The input image is pre-processed before inputting the proposed system. Light correction and a histogram equalizer were used to equalize the intensity values in each window.

The second stage focuses on merging overlapping detections and the arbitration process. The same face is detected many times with adjacent locations, the centre of these locations being considered as the centre of the detected face, and if two face locations are overlapped, the one with the highest score is considered the face location. Multiple networks were used to improve detection accuracy by ANDing (or ORing) the output of two networks over different scales and positions. Rowley's system was evaluated using 130 images containing 507 faces, the images having been collected from newspaper pictures, photographs and the World Wide Web. To train the system on false examples, 1000 images with random pixel intensities were generated. The detection rate of this system ranged from 78.9% - 90.5% depending on the arbitration used (ANDing or ORing).

Rowley's scheme is tested on our video database and detected al l the faces in the videos, and therefore was used for the purpose of this study.

4. Lip detection

Over the last few decades, the number of applications that are concerned with the automatic processing/analysis of human faces has grown remarkably. Many of these applications have a particular interest in the lips and mouth area. For such applications a robust and real-time lips detection/localization method is a major factor contributing to their reliability and success. Since lips are the most deformable part of the face, detecting them is a nontrivial problem, adding to the long list of factors that adversely affect the performance of image processing/analysis schemes, such as variations in lighting conditions, pose, head rotation, facial expressions and scaling.

The lips and mouth region are the visual parts of the human speech production system; these parts hold the most visual speech information, therefore it is imperative for any VSR system to detect/localize such regions to capture the related visual information, i.e. we cannot read lips without seeing them first. Therefore, lip localization is an essential process for any VSR system.

4.1 Existing trends in lip detection

Many techniques for lips detection/localization in digital images have been reported in the literature, and can be categorized into two main types of solutions:

1. *Model-based* lips detection methods. Such models include "Snakes", Active Shape Models (ASM), Active Appearance Models (AAM), and deformable templates.
2. *Image-based* lips detection methods. These include the use of spatial information, pixel colour and intensity, lines, corners, edges, and motion.

4.1.1 Model-based lip detection methods

This approach depends on building lip model(s), with or without using training face images and subsequently using the defined model to search for the lips in any freshly input image. The best fit to the model, with respect to some prescribed criteria, is declared to be the location of the detected lips. For more about these methods see (Cootes & Taylor, 1992; Kass, et. al., 1987; Yuille, et. al, 1989).

4.1.2 Image-based lip detection methods

Since there is a difference between the colour of lips and the colour of the face region around the lips, detecting lips using colour information attracted researchers' interest recently because of simplicity, not being time consuming, and the use of fewer resources, e.g. low memory, allowing many promising methods for lip detection using colour information to emerge. The most important information that researchers focus on include the red and the green colours in the RGB colour system, the hue of the HSV colour system, and the component of the red and blue in the YCbCr colour system. Some researchers used more information from the lip edges and lip motion. A well known mixed approach is called the "hybrid edge" (Eveno, et. al., 2002).

4.2 The adopted lip detection method

Among the model-based lip-detection methods, the active shape models are the most common and best performing technique for lip detection. However, this approach is basically affected by factors such as facial hair and does not meet some of the functional requirements (e.g. in terms of speed). "The implementation of Active Shape Model ASM was always difficult to run in Real-Time" (Guitarte, et. al., 2003). The same thing applies to the AAM approach. In fact, AAM is slower than ASM.

In grey level images and under diffuse lighting conditions, the external border of the lip is not sharp enough (see Figure 2a), and this makes the use of techniques based on the information provided by these images alone ineffective (Coianiz, et. al., 1996). Moreover, the ASM and the AAM algorithms are sensitive to the initialization process. When initialization is far from the target object, they can converge to local minima (see Figure 2a).

Other problems, like the appearance of the moustache, beard, and accessories, contribute to the problems and challenges that model-based lip detection methods undergo. Figure 2b illustrates some of these problems.

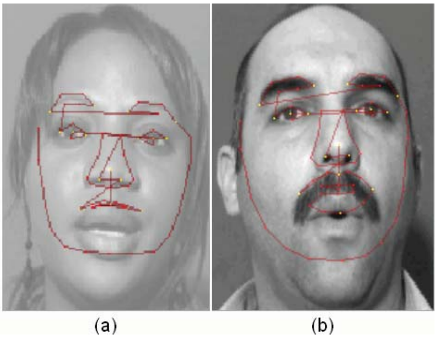
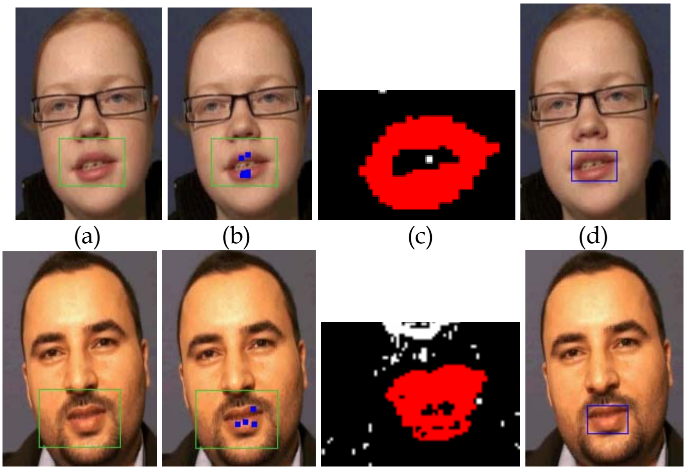


Fig. 2. Facial features detection using MASM*, (a) lip detection converges to local minima, (b) the effect of facial hair on ASM convergence.

Since the VSR problem needs several pre-processing steps, e.g. face and lips detection, it is vital for the VSR system to have faster solutions for these steps in order that the final solution can work in real time.



(a) (b) (c) (d)

Fig. 3. The different stages of the "nearest colour", a) face detection followed by ROI defining, b) initial clustering using the YCbCr, c) binary image of ROI resulting from the nearest colour algorithm, d) final lip detecting.

[*] "MSAM" is a state-of-art ASM free library, developed by (Milborrow & Nicolls, 2008), and can be downloaded from the following link: http://www.milbo.users.sonic.net/stasm/download.html

In order to overcome the above-mentioned difficulties, we shall use a colour-based method (for lip localization in our study), in spite of being vulnerable to variations in light conditions. Such an effective method is described in our previous work (Hasanat, 2009). This method is based on using the YCbCr approach to find at least any part of the lip as an initial step. Then we use all the available information about the segmented lip-pixels such as r, g, b, warped hue, etc. to segment the rest of the lip. The mean is calculated for each value, then for each pixel in ROI, and Euclidian distance from the mean vector is calculated. Pixels with smaller distances are further clustered as lip pixels. Thus, the rest of the pixels in ROI will be clustered (to lip/non-lip pixel) depending on their distances from the mean vector of the initial segmented lip region. See Figure 3.

The method was evaluated on 780,000 frames of the in-house database; the experiments show that the method localizes the lips efficiently, with high level of accuracy (91.15%).

5. Features extraction and recognition

VSR systems require the analysis of feature vectors, which are extracted from the speech-related visual signals, in ROI in the sequence of the speaker face frames while uttering the spoken word/speech. Ideally, the required feature representations of words must capture specific visual information that is closely associated with the spoken word, to enable the recognition of the word and distinguish it from other words. Unlike the visemic approach, the visual words technique depends on finding a signature for the whole word, instead of recognizing each part (Viseme) of the word alone. To find such a signature, or a signal for each word, we need to find a proper way of extracting the most relevant features, which play an important role in recognizing that word.

An appearance-based approach to visual speech feature extraction ignores the fact that mouth appearance varies from one person to another (even when two persons speak the same word). Thus, using the appearance-based feature extraction alone does not take individual differences into consideration, and leads to inaccurate results. Moreover, appearance-based feature extraction methods mostly lack robustness in certain illumination and lighting conditions (Jun & Hua, 2009).

In this chapter, we adopt the hybrid-based approach, and we expand on the list of features beyond traditionally adopted ones such as the height and width of the speaker's lips. Indeed, there is valuable information encapsulated within the ROI that has a significant association with the spoken word, e.g. the appearance of the tongue and teeth in the image during the speech. The appearance of the teeth (for instance) occurs while uttering specific phonemes (the dentals and labio–dentals). At the same time, focusing only on the image-based features (appearance and transformed-based features) yields image-specific features, and it is sometimes difficult to generalize about those features on other videos or speakers. These results are backed up by (Jun & Hua, 2009).

The visual signal associated with a phoneme is rather short and hence their visual features are extracted from "representative" image frames. However, the visual signals associated with words are of longer duration involving tens of frames that vary in many ways. Hence the need to supplement/modify the set of features used in a visemic system by including some features relating to variation of frames along the temporal axis. There are many ways to represent such features, but we shall include two seemingly obvious features: an image quality parameter that measures the deviation/distortion of any frame from its predecessor,

as well as the amount of mutual information between a frame and its predecessor. Such features are expected to compensate for the fact that many words do share some phonemes.
The list of features adopted in this chapter is by no means exclusive, but it was limited out of a desire to minimize the number for efficiency purposes and to a manageable set of features for which their impact on the accuracy of the intended VSR system can be estimated experimentally. The following is the proposed list of features that will be extracted from the sequences of the ROIs of the mouth areas during the uttering of the word (see Figure 4):

1. The height (H) and width (W) of the mouth, i.e. ROI height and width (geometric-based features).
2. The mutual information (M) between consecutive frames ROI in the discrete wavelet transform (DWT) domain (image-transformed-based features based on temporal information).
3. The image quality value (Q) of the current ROI with reference to its predecessor measured in the DWT domain (image-transformed-based features based on temporal information).
4. The ratio of vertical to horizontal features (R) taken from DWT of ROI (image-transformed-based features based on temporal information).
5. The ratio of vertical edges to horizontal edges (ER) of ROI (image-transformed-based features).
6. The amount of red colour (RC) in ROI as an indicator of the appearance of the tongue (image-appearance-based features).
7. The amount of visible teeth (T) in the ROI (image-appearance-based features).

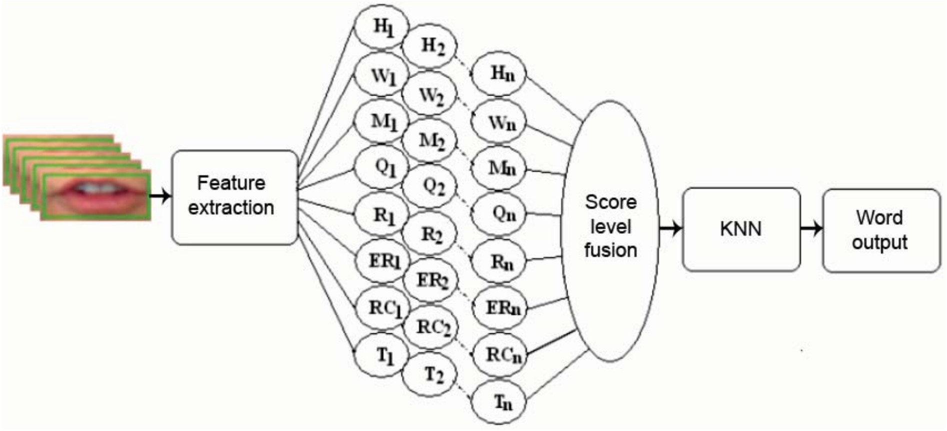


Fig. 4. The proposed feature extraction and recognition method.

As can be noticed from Figure 4, for each spoken word, eight feature vectors of length *n* (number of frames) are extracted, forming 8 different signals. This feature extraction method produces 8 signals for each uttered word, creating 8-dimensional feature space. Those signals maintain the dynamic of the spoken word, which contains a good portion of information; on the contrary, the visemic approach does not take into consideration the dynamic movement of the mouth and lips to produce a spoken word (Yu, 2008).

Accordingly, for each word we would extract a time-series of 8-dimensional vectors. The main difficulties in analysing of these time-series stem from the fact that their lengths not only differ between the spoken words, but also differ between different speakers uttering the same word and between the different occasions when the same word is uttered by the same speaker. In what follows we describe each of the 8 features. We assume that for each frame of a video, the speaker's face and lips are first localized (see sections 3 and 4) to determine the ROI from which these features are extracted (see Figure 5).

5.1 The height and width features of the mouth

Some VSR studies used lip contour points as shape features to recognise speech. For example, Wang et al. (2004) used a parameter set of a 14 points ASM lip to describe the outer lip contour. In addition, Sugahara et al. (2004) employed a sampled active contour model (SACM) to extract lip shapes. Determining the exact lips contour is rather problematic due to the little differences in the pixel values between the face and the lips. Here, we argue that it is not necessary (redundant) to use all or some of the lip's contour points to define the outer shape of the lips, where the height and width of the mouth backed up with a bounding ellipse is enough to approximate the real outer contour of the lips (see Figure 5).

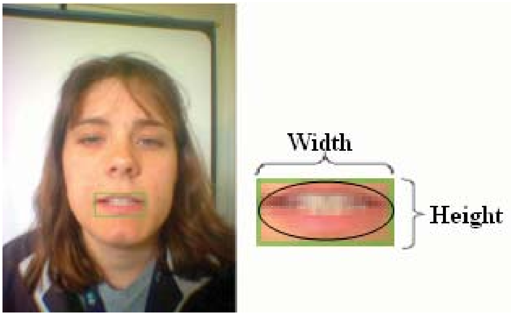


Fig. 5. Lips geometric feature extraction; width and height.

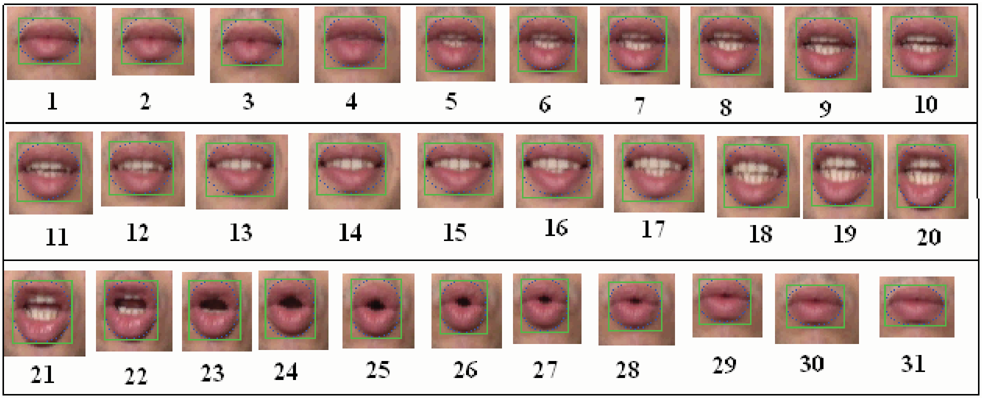


Fig. 6. The change of the mouth shape while uttering the word "Zero", the blue dotted ellipse shows the approximated lip contour using the ellipse assumption.

In our proposal, the height and width of the mouth area are determined by an approximation of the minimal rectangular box that contains the mouth area. The corners of the detected mouth box will be eliminated using the assumption that the mouth shape is the largest ellipse inside the minimal box. This assumption also helps in reducing redundant data when extracting the other features in the scheme. Sometimes lips are not horizontally symmetric due to different ways of speaking, the ellipse assumption forces such symmetries, and alleviates such differences between individuals (see Figure 6). These changes in height and width of the mouth create two signals (W and H) that represent changes during the time of uttering a specific word.

5.2 The mutual information (M) feature

Mutual information between 2 random variables X and Y defines the dependency of these variables, i.e. mutual information reveals how much X contains information about Y, and vice versa. Mutual information can be utilized to quantify the temporal correlation between frames of a video sequence, so it can calculate the amount of redundancy between any two frames. The temporal change of the appearance of ROI is caused by uttering a new/different phoneme. For example, the mouth appearance will change while switching from phoneme [ē] to phoneme [d] when uttering the word "feed". Therefore, it is sensible to use the mutual information to measure some aspects of the change in the mouth area between consecutive ROIs. The mutual information M between two random variables X and Y is defined by:

$$M(X;Y) = \sum_{x}\sum_{y} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{1}$$

where p(x,y) is the joint probability mass function (PMF) of random variables X and Y (in our case mouth image (ROI) in current frame X, and previous mouth image in frame Y), p(x) and p(y) represent the marginal PMF of X and Y respectively. To use the mutual information formula, the size of both of the random variables must be the same, but because the height and width of ROI are changing over time while uttering different phonemes, consecutive ROIs might not be of the same size. To solve this problem, both ROIs are scaled to a predefined size, say 50 x 50 pixels.

Computing the mutual information in the spatial domain is inefficient and is influenced by many factors including the presence of noise and variation in lighting conditions. Instead, measuring the mutual information in the frequency domain provides a more informative mechanism to model changes between successive ROIs in different frequency sub-bands. Here we apply the DWT on both the current and the previous ROI. The mutual information formula is applied 4 times, one for each wavelet sub-band, and the average of the four values is taken as the mutual information feature for that frame or ROI (see Figure 7). Transforming both ROIs into the wavelet domain helps to reduce the effect of noise and variation in lighting conditions. For simplicity and efficiency, the DWT decomposition of the ROIs is implemented using the Haar filter.

5.3 The quality measure (Q) feature

There are many image quality measures proposed in the literature. Most of them attempt to find the amount of distortion in one image by referring to another image. Unlike the mutual information measurement, which attempts to measure the amount of dependency or similarity between two images, quality measures attempt to measure how different one image is from another.

Thinking again of the consecutive ROIs, a quality measure between them can tell something about change/distortion occurring due to an uttered phoneme. Therefore, any distortion in the current ROI, as compared to the previous ROI, is an indicator of changes in the structure of the mouth region. The amount of distortion can be measured by a quantitative quality measure, and considered as a feature at that frame or ROI.

This study utilizes a universal image quality index proposed by (Wang & Bovik , 2002) because it is a fast mathematical quality measure, and models image distortion as a combination of loss of correlation, luminance distortion, and contrast distortion. The quality measure Q is given by:

$$Q = \frac{4\sigma_{xy}\bar{x}\,\bar{y}}{(\sigma_{\mathrm{x}}^2 + \sigma_{\mathrm{y}}^2)[(\bar{x})^2 + (\bar{y})^2]} \tag{2}$$

where Q Є [-1,1],

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad , \quad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \quad , \quad \sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2 \quad ,$$

$$\sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2 \quad , and \quad \sigma_{xy}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})$$

The best value for Q is when there is no distortion in the current ROI compared to the previous ROI; the value then is equal to 1 or -1 and the maximum distortion is measured when Q = 0.

This formula is very sensitive to luminance, because it models image distortion as luminance distortion, as well as loss of correlation and contrast distortion. This problem is solved by using the same approach that was used for the mutual information feature, i.e. using the DWT decomposed ROIs. Again, 4 quality measures (Q) are computed, one for each wavelet sub-band (the HH, HL, LH, and the LL). Then the average of the four values is taken as the quality measure feature for that frame or ROI (see Figure 7). For compatibility, we also use the Haar filter and then both ROIs are scaled to 50 x 50 pixels. The average of both the mutual M, and the quality Q features is defined by:

$$M_i = \frac{M(LL_i;\mathrm{LL}_{i\ 1}) + M(HL_i;HL_{i\ 1}) + M(LH_i;LH_{i\ 1}) + M(HH_i;HH_{i\ 1})}{4} \tag{3}$$

$$Q_i = \frac{Q(LL_i;LL_{i\ 1}) + Q(HL_i;HL_{i\ 1}) + Q(LH_i;LH_{i\ 1}) + Q(HH_i;HH_{i\ 1})}{4} \tag{4}$$

where M_i and Q_i are the mutual and quality features at frame *i* respectively, LL_i, HL_i, LH_i and HH_i are the wavelet sub-bands of the current ROI, and LL_{i-1}, HL_{i-1}, LH_{i-1} and HH_{i-1} are the wavelet sub-bands of the previous ROI.

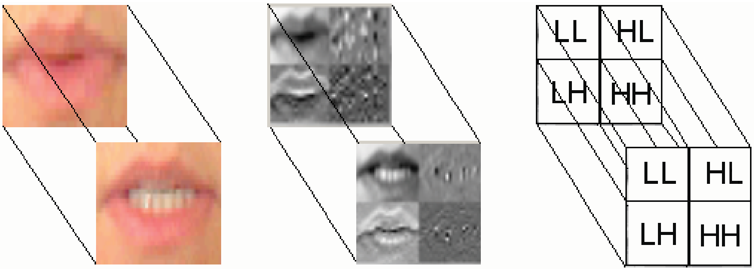


Fig. 7. (1st row) The previous mouth and its Haar wavelet, (2nd row) Current mouth and its Haar wavelet.

5.4 The ratio of vertical to horizontal features (R)

The DWT of an image I, using any wavelet filter, then histogram of the approximation sub-band LL approximates that of the original image while the coefficients in each of the three other sub-bands have a Laplacian distribution with 0 means (Al-Jawad, 2009). Moreover, in each non-LL-sub-band the further away from the mean a coefficient is, the more likely it is associated with a significant image feature such as edges/corners.

Here we adopt the above approach to identify feature-related pixels as the significant coefficients in the Non-LL sub-bands, i.e. the feature points are the ones with values greater than (median + standard deviation), and less than (median – standard deviation). The ratio (*R*) of the vertical features obtained from wavelet sub-band HL to the number of the horizontal ones gained from the LH is given by:

$$R = \frac{V}{H} \tag{5}$$

where V = number of vertical features, and H = number of horizontal features. Accordingly, by substituting V and H in equation 5, we get equation 6.

$$R = \frac{\sum_x \sum_y \begin{cases} 1 & (HL_{median} + \sigma_{HL}) \le HL(x,y) \le (HL_{median} + \sigma_{HL}) \\ 0 & otherwise \end{cases}}{\sum_x \sum_y \begin{cases} 1 & (LH_{median} + \sigma_{LH}) \le LH(x,y) \le (LH_{median} + \sigma_{HL}) \\ 0 & otherwise \end{cases}} \tag{6}$$

where HL_{median} and LH_{median} are the medians of the wavelet sub-band *HL* and *LH* respectively, *HL(x,y)* and *LH(x,y)* the intensity value at location (x,y) in both *HL* and *LH* wavelet sub-bands, σ_{HL} and σ_{LH} are the standard deviation in both of the mentioned sub-bands. Figure 8 demonstrates the correlation between the mouth appearance and its ratio (*R)* property while speaking.

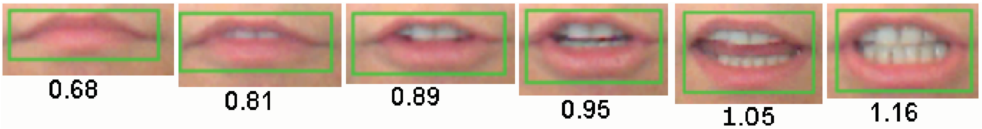


Fig. 8. The co-relation between the mouth appearance and its ratio (R).

As can be seen from Figure 8, the ratio R is high when there are a lot of vertical features of ROI compared to the horizontal ones (the mouths on the right), and R is low when vertical features are low and/or horizontal features are high (the mouths on the left).

5.5 The ratio of vertical edges to horizontal edges (ER)

The ratio of vertical edges to horizontal edges (ER) of ROI is obtained by using the Sobel edge detector. The summation of the absolute values of the vertical filter demonstrates the amount of vertical edges in the ROI. In addition, the summation of the absolute values of the horizontal filter demonstrates the amount of horizontal edges in the ROI. The ratio of the vertical edges to the horizontal ones is given by:

$$ER = \frac{\sum_{x=1}^{W}\sum_{y=1}^{H}\sum_{i=1}^{1}\sum_{j=1}^{1}\left|ROI(x+i,y+j)(S_v(i+1,j+1))\right|}{\sum_{x=1}^{W}\sum_{y=1}^{H}\sum_{i=1}^{1}\sum_{j=1}^{1}\left|ROI(x+i,y+j)(S_h(i+1,j+1))\right|} \tag{7}$$

where ROI(x,y) is the intensity value at the location (x,y) of the mouth region, W is the width of ROI, *H* is the height of ROI. *Sv*, and *Sh* are Sobel vertical and horizontal filters respectively.

When the mouth is stretched horizontally, the amount of horizontal edges increases, so ER decreases. When the mouth is opened, the amount of vertical edges tends to increase, and this increases the ER. Therefore, ER reveals something about the appearance of the mouth at a particular time.

5.6 The appearance of the tongue (RC)

Some phonemes like [*th*] involve the appearance of the tongue, i.e. moving the tongue and showing it helps to utter such phonemes. Therefore detecting the tongue in the ROI reveals something about the uttered phoneme and, by implication, the visual word.

However, it is difficult to model the tongue; the only available cue is its red colour. Therefore, the amount of red colour (RC) in the ROI will be taken to represent the appearance of the tongue, as well as the lip colour. Since the lip is captured within the ROI, the change of the red colour amount is then a cue for the appearance of the tongue. The different size of the tongue and lip from person to person is not problematic, hence all the features are scaled to the range [0,1], and the ratio of the red colour to the size of ROI is considered. This ratio can be calculated using the following equation:

$$RC = \frac{\sum_{x1}^{W}\sum_{y=1}^{H} red(ROI(x,y))}{(W)(H)} \tag{8}$$

where *red*(*ROI*(x,y)) is the red component value of the RGB colour system at the (x,y) of the mouth region, W is the width of ROI, H is the height of ROI.

5.7 The appearance of the teeth (T)

Some phonemes like [s] incorporate the appearance of the teeth, i.e. showing teeth helps to utter such phonemes. Therefore detecting teeth in the ROI is a visual cue for uttering such phonemes and enriches the visual words signatures. The major characteristic that distinguishes teeth from other parts of the ROI is the low saturation and high intensity values (Goecke, 2000). By converting the pixels values of ROI to 1976 CIELAB colour space (L*, a*, b*) and 1976 CIELUV colour space (L*, u*, v*), the teeth pixel has a lower a* and u* value than other lip pixels (Liew et. al., 2003). A teeth pixel can be defined by:

$$t = \begin{cases} 1 \; a^{*} \leq (\mu_a - \sigma_a) \\ 1 \; u^{*} \leq (\mu_u - \sigma_u) \\ 0 \; otherwise \end{cases} \tag{9}$$

where $\mu_a, \sigma_a \; and \; \mu_u, \sigma_u$ are the mean and standard deviation of a* and u* in ROI respectively. The appearance of the teeth can be defined by the number of teeth pixels in ROI. Therefore, the amount of teeth in ROI is given by:

$$T = \sum_{x=1}^{W} \sum_{y=1}^{H} t(x,y) \tag{10}$$

5.8 The classification process

All the previous features are normalized to the range [0,1] to alleviate the individual differences, and different scales of mouth caused by different distances from the camera, i.e. the different sizes of ROIs. For each property, a feature vector (a signal) is obtained to represent the spoken word from that feature perspective. Consequently, for each spoken word we get a feature matrix. The feature matrix has a fixed number of columns, but with a different number of rows, depending on the uttered word, and on the different speed of uttering words.

To compare signals with different lengths, we use the Dynamic Time Warping (DTW) method, and linear interpolation. For the fusion of the aforementioned features, we used score level fusion, which includes the use of each feature vector alone, using an empirical weighting technique to give different weights for the features, to capture the reliability of each feature vector, depending on how informative they are.

For each signal, the distances are measured with other signals from the training data, using DTW or Euclidian distance after linear interpolation, to overcome the different signal lengths. According to the K-Nearest-neighbour (KNN), the minimum k weighted averages are considered to predict the class (word) by announcing the maximum occurrence class in the nearest k as the predicted class.

6. Some experimental results

We evaluate the discussed VSR system using our in-house video database, in addition to the following main types of experiments:

1. **Speaker-dependent experiment:** this was conducted on each subject alone, all the test examples, and the training examples pertaining to the same subject (person). The main goal of this experiment is to test the way of speaking unique to each person, and each one's ability to produce a visual signal that was easily read. These experiments use leave-one example-out cross-validation protocol, test samples came from session 1 and training samples from session 2 for the same subject.
2. **Speaker-independent experiment:** In this type of experiment, the computer tests each subject against the rest of the subjects. Each time, one subject is taken out of the training set and is tested against the remaining subjects in the database. The training set does not contain any examples belonging to the tested subject. So the leave-one-subject-out cross-validation is used to evaluate the system. This type of experiment neglects the individual differences in appearance, and in the way of speaking.

Subject	**Speaker dependent**	**Speaker independent**	**Subject**	**Speaker dependent**	**Speaker independent**
Female01	**69%**	27%	Male04	**63%**	16%
Female02	**97%**	49%	Male05	**85%**	36%
Female03	**87%**	35%	Male06	**65%**	19%
Female04	**81%**	39%	Male07	**84%**	36%
Female05	**83%**	26%	Male08	**75%**	27%
Female06	**75%**	43%	Male09	**92%**	41%
Female07	**82%**	31%	Male10	**88%**	31%
Female08	**85%**	29%	Male11	**84%**	53%
Female09	**81%**	41%	Male12	**69%**	26%
Female10	**88%**	42%	Male13	**53%**	15%
Male01	**79%**	43%	Male14	**69%**	33%
Male02	**61%**	15%	Male15	**83%**	39%
Male03	**44%**	23%	Male16	**62%**	28%
	All			**76.38%**	33%
	Females			83%	36%
	Males			72%	30%
	Excluding moustache & beard			77%	33%
	Moustache & beard			71%	30%

Table 2. VSR system Word recognition rates

It can be noticed from Table (2) that the overall WER of speaker-dependent experiments was (76.38%), and it was only 33% for the speaker-independent experiment. Our experiments show that the speaker-dependent word recognition rate is much higher than that of the speaker-independent; this claim is backed up by several researchers such as (Jun & Hua, 2009). Individual differences in the mouth appearance, and in the way of talking, produce different visual and audio signals for the same spoken word, which emphasizes that the visual speech recognition problem is a **speaker-dependent problem**.

We can notice also the negative effect of the facial hair on the results, when excluding subjects with moustaches and beards performance increased by 6% (from 71% to 77%). This explains the female's best results (83%). Moreover, the training set contains native and non-native speaker subjects, and each of the non-native speakers has his/her own way of uttering English words, for example the word "determine" is pronounced in 3 different ways by the non-native subjects, "di-tur-min", "de-teir-main" and "de-ter-men". This gives the training set different signatures for the same word, which confuses the recognition algorithm and contributes to the "bad examples" pool*. Moreover, the training set contains different ethnic groups, African, Asian, Middle Eastern and European; these groups are different in appearance. Furthermore, there are also differences in the appearance of males versus females, and the differences between age groups, i.e. different colours and shapes of the lips and mouth region. This variety leads to different features being extracted from the same word, which again contributes to the "bad examples" pool and leads to unexpected results.

Another interesting observation is the large difference between the individual ability to produce the visual signal while talking (word recognition rate varies from 44% to 97%). We found that some participants have less ability to produce this signal, i.e. they talk with minimum lip movement, which makes it a difficult task to read their lips, even by human intelligence. We termed those persons "visual-speechless persons" (VSP). In our experiments, we found that Male02 and Male13 are VSP (see Figure 9).

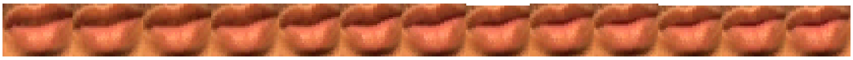
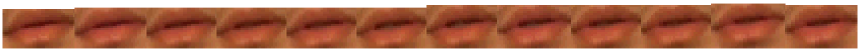
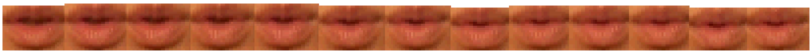
Subject	**Visual representation**
Male02	
Male13	
Female2	

Fig. 9. Illustrating VSP concept, 1st and 2nd rows show the appearance of the word "two" uttered by two VSPs and the 3rd row shows the same word uttered by a normal person.

The previous Figure (9) shows that the VSPs do not produce clear visual signals. i.e. the appearance, shape and dynamic of ROI from the 1st to the last frame, seems to be the same (unchanged to some extent), which makes it difficult to produce a unique signature for their visual speech, resulting in low WER for such subjects.

7. Chapter summary and conclusion

In this chapter, we described a complete VSR system, which includes face and lip detection/localization, features extraction and recognition. We evaluated the described scheme using two types of experiments, speaker-dependent and speaker-independent.

* Some examples in the training set, which are meant to represent some words, are closer to other words, e.g. outliers.

These experiments were carried out using a special database, which was designed for evaluation purposes. There were high word recognition rates for some subjects, and low ones for some others. Several reasons were found that affected the various results such as the appearance of facial hair, and the individual's aptitude to produce a clear visual signal. Some subjects produce weak signals (termed as VSP).

Results of the speaker dependents experiments were much better than that of the speaker independent. Therefore, we consider the VSR as speaker dependent problem, and to confirm such a result we need to further investigate VSR using different databases, and try to find some appearance invariant features, to minimize the effect of the visual appearance differences between individuals.

The major challenge for VSR is the lack of information in the visual domain, compared to the audio domain, perhaps because humans have yet to evolve to have need of a more sophisticated communication system. For example, it was sufficient for man's survival to use sound to warn friends if there was an enemy or a predator around without having to see them. Therefore, humans did not worry about producing a sufficient visual signal while talking. This major challenge, along with some others, opens the door for more research in the future, to compensate for the lack of information.

8. Acknowledgment

The author would like to acknowledge the financial support of Mutah university/Jordan (www.Mutah.edu.jo). In addition to professor Sabah Jassim for all his viable advices and discussions.

9. References

Al-Jawad, N., (2009). *Exploiting Statistical Properties of Wavelet Coefficients for Image/ Video Processing and Analysis Tasks*, PhD thesis, University of Buckingham, UK.

Arsic, I., & Thiran, J. (2006). Mutual Information Eigenlips For Audio-Visual Speech Recognition, *Proceedings of the 14th European Signal Processing Conference (EUSIPCO).*

Belongie, S., & Weber, M. (1995). Recognising Spoken Words from Lip Movement, *Technical Report CNS/EE248*, California Institute of Technology, USA.

Bregler, C., & Konig, Y. (1994). Eigenlips for Robust Speech Recognition, *Proceedings of ICASSP94*, Vol. II, Adelaide, Australia, 669-672.

Bruhn, M. E. (1920). *The Muller Walle Method of Lip Reading for the Deaf*, press of Thos. P. Nichols & Son Co. Lynn, Ma. USA.

Coianiz, T., Torresani, L., & Caprile, B., (1996). 2D deformable models for visual speech analysis, in DG Stork & ME Hennecke (Eds.), *Speechreading by Humans and Machines*, pp. 391-398.

Cootes, T. F. & Taylor, C. J., (1992). Active Shape Models - Smart Snakes, *Proceedings of the British Machine Vision Conference*, Springer-Verlag, pp. 266-275.

De Land F. (1931). *The Story of Lip Reading*, The Volta Bureau, Washington D. C., USA.

Eveno, N., Caplier, A., & Coulon, P. Y. (2002). A Parametric Model for Realistic Lip Segmentation, *Proceedings of ICARCV 02*, IEEE Press 3, pp. 1426–1431.

Garcia, C., & Tziritas, G., (1999). Face Detection Using Quantized Skin Color Regions Merging and Wavelet Packet Analysis, *IEEE Transactions On Multimedia*, Vol.1, No. 3, pp. 264-277.

Goecke, R., Millar, J.B., Zelinsky, A., & Robert-Ribes, J. (2000) Automatic extraction of lip feature points, *Proceedings of the Australian Conference on Robotics and Automation*, pp. 31-36.

Guitarte, J., Lukas, K., & Frangi, A.F. (2003). Low Resource Lip Finding and Tracking Algorithm for Embedded Devices, *ISCA Tutorial and Research Workshop on Audio Visual Speech Processing*, pp. 111-116.

Gurban, M. & Thiran, J. (2005). Audio-Visual Speech Recognition With A Hybrid Svm-Hmm System, *Proceedings of the 13th European Signal Processing Conference (EUSIPCO).*

Hassanat, A. B.A, (2009). *Visual Words for Automatic Lip-Reading*, PhD thesis, University of Buckingham, UK.

Hazen, T. J., Saenko, K., La, C.H., & Glass, J., (2004) A segment-based audio-visual speech recognizer: Data collection, development and initial experiments, *Proceedings of the International Conference on Multimodal Interfaces*, pp. 235-242.

Hjelmas, E., & Low, B.K., (2001). Face Detection: A Survey, *Computer Vision and Image Understanding*, vol. 3, pp. 236-274.

Jun, H. & Hua, Z. (2009). Research on Visual Speech Feature Extraction, *Proceedings of the International Conference on Computer Engineering and Technology*, Volume 2, pp. 499 – 502.

Kass, M., Witkin, A., & Terzopoulos, D. (1987). Snakes: Active contour models, *International Journal of Computer Vision*, volume 1, pp. 321-33.

Leszczynski, M., & Skarbek, W. (2005). Viseme Recognition – A Comparative Study, *Proceedings of IEEE International Conference on Advanced Video and Signal-Based Surveillance.*

Liew, A.W.C., Leung, S.H., & Lau, W.H., (2003). Segmentation of Color Lip Images by Spatial Fuzzy Clustering, *IEEE Transactions on Fuzzy Systems*, vol.11, no.4, pp. 542-549.

Lucey, P., & Sridharan, S. (2008). A Visual Front-End for a Continuous Pose-Invariant Lip-reading System, *Proceedings of the 2nd International Conference on Signal Processing and Communication Systems*, 15-17 December 2008, Australia, Queensland, Gold Coast.

Morris C., Koreman j., Sellahewa h., Ehlers j., Jassim s., Allano L., & Garcia-salicetti s. (2006). The SecurePhone PDA Database, Experimental Protocol and Automatic Test Procedure for Multimodal User Authentication, *technical report, Secure-Phone (1EC IST-2002-506883) project*, Version 2.1.

Neti, C., Potamianos, G. & Luettin, J. (2000). Audio-visual speech recognition, *Final Workshop 2000 Report, Center for Language and Speech Processing*, The Johns Hopkins University, Baltimore, MD.

Osuna, E., Freund, R., & Girosi, F. (1997). Training support vector machines: An application to face detection. *Proceedings of CVPR*, pp. 130–136.

Petajan, E. (1984). *Automatic lipreading to enhance speech recognition*, Ph.D. Dissertation, University of Illinois at Urbana-Champaign, USA.

Potamianos G., Neti C., Iyengar G., & Helmuth E. (2001). Large-Vocabulary Audio-Visual Speech Recognition by Machines and Humans, *Proceedings of EUROSPEECH,* pp. 1027-1030, Aalborg, Denmark, 2001.

Potamianos, G., Neti, C., Gravier, G., Garg, A., & Senior, A. W. (2003). Recent Advances in the Automatic Recognition of Audio-Visual Speech, Invited, *Proceedings of the IEEE*, vol. 91, no. 9, pp. 1306-1326.

Romdhani, S., Torr, P., Lkopf, B., & Blake, A. (2004). Efficient Face Detection by a Cascaded Support Vector Machine Expansion, *Royal Society of London Proceedings Series A*, vol. 460, pp. 3283-3297.

Rowley, H., Baluja, S.,& Kanade, T. (1998). Neural network-based face detection, *PAMI 20*, pp. 23–38.

Saenko, K., Livescu, K., Glass, J. & Darrell, T. (2005). Production Domain Modeling Of Pronunciation For Visual Speech Recognition, *Proceedings of ICASSP*, Philadelphia.

Sagheer, A., Tsuruta, N., Taniguchi, R. I. & Maeda, S. (2006). Appearance feature extraction versus image transform-based approach for visual speech recognition, *International Journal of Computational Intelligence and Applications*, Vol. 6, pp. 101–122.

Samaria, F.S., (1994). *Face Recognition Using Hidden Markov Models*, PhD thesis, Univ. of Cambridge, UK.

Schneiderman, H., & Kanade, T. (2000). A statistical method for 3d object detection applied to face and cars, *Proceedings of CVPR*, pp. 746–751.

Sung, K-K., & Poggio, T., (1998). Example-Based Learning for View-Based Human Face Detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, pp. 39-51.

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition, *Journal of Cognitive Neurosci*, vol. 3. no.1, pp. 71-86.

Viola, P. & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features, *Proceedings of the International Conference on Computer Vision and Pattern Recognition – CVPR*, Kauai, HI, USA, pp. 511–518.

Wang, Z., & Bovik, AC. (2002). A universal image quality index, *IEEE Signal Process Lett. 9*, pp. 81–84.

Wang, S.L., Lou, W.H., Leung, S.H. & Yan, H. (2004). A real-time automatic lip-reading system, *Proceedings of the IEEE International Symposium on Circuits and Systems*, Vancouver BC, Canada, vol. 5, pp. 101-104.

Werda, S., Mahdi, W., & Ben-Hamadou, A. (2007). Lip Localization and Viseme Classification for Visual Speech Recognition, *International Journal of Computing & Information Sciences*, Vol.5, No.1.

Yang, M.H., Kriegman, D., & Ahuja, N. (2002). Detecting Faces in Images: A Survey, *IEEE Trans. Pattern Anal. Mach. Intell.*, Volume 24, pp. 34 – 58.

Yau, W., Kumar, D., & Arjunan, S. (2006). Voiceless Speech Recognition Using Dynamic Visual Speech Features, *HCSNet Workshop on the Use of Vision in Human-Computer Interaction, (VisHCI 2006)*, pp. 93-101.

Yu, D. (2008). *The Application of Manifold based Visual Speech Units for Visual Speech Recognition*, PhD thesis, Dublin City University, Dublin, Ireland.

Yuille, A., Cohen, D. S. & Hallinan, P. W., (1989). Feature extraction from faces using deformable templates, *Proceedings of the IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn*, pp. 104-109.

15

Towards Augmentative Speech Communication

Panikos Heracleous[1], Denis Beautemps[2], Hiroshi Ishiguro[3],
and Norihiro Hagita[4]
[1,3,4]*ATR, Intelligent Robotics and Communication Laboratories Japan Science and Technology Agency, CREST*
[2]*GIPSA-lab, Speech and Cognition Department, CNRS-Grenoble University*
[1,3,4]*Japan,*
[2]*France*

1. Introduction

Speech is the most natural form of communication for human beings and is often described as a unimodal communication channel. However, it is well known that speech is multimodal in nature and includes the auditive, visual, and tactile modalities. Other less natural modalities such as electromyographic signal, invisible articulator display, or brain electrical activity or electromagnetic activity can also be considered. Therefore, in situations where audio speech is not available or is corrupted because of disability or adverse environmental condition, people may resort to alternative methods such as augmented speech.

In several automatic speech recognition systems, visual information from lips/mouth and facial movements has been used in combination with audio signals. In such cases, visual information is used to complement the audio information to improve the system's robustness against acoustic noise (Potamianos et al., 2003).

For the orally educated deaf or hearing-impaired people, lip reading remains a crucial speech modality, though it is not sufficient to achieve full communication. Therefore, in 1967, Cornett developed the Cued Speech system as a supplement to lip reading (O.Cornett, 1967). Recently, studies have been presented on automatic Cued Speech recognition using hand gestures in combination with lip/mouth information (Heracleous et al., 2009).

Several other studies have been introduced that deal with the problem of alternative speech communication based on speech modalities other than audio speech. A method for communication based on inaudible speech received through body tissues has been introduced using the Non-Audible Murmur (NAM) microphone. NAM microphones have been used for receiving and automatically recognizing sounds of speech-impaired people, for ensuring privacy in communication, and for achieving robustness against noise (Heracleous et al., 2007; Nakamura et al., 2008). Aside from automatic recognition of NAM speech, silicon NAM microphones were used for NAM-to-speech conversion (Toda & Shikano, 2005; Tran et al., 2008).

A few researchers have addressed the problem of augmented speech based on the activation signal of the muscles produced during speech production (Jou et al., 2006). The OUISPER project (Hueber et al., 2008) attempts to automatically recognize and resynthesize speech based on the signals of tongue movements captured by an ultrasound device in combination with lip information.

In this article, automatic recognition of Cued Speech for French and Non-Audible Murmur (NAM) recognition are introduced. Cued Speech is a visual mode for communication in the deaf society. Using only visual information produced by lip movements and hand shapes, all the sounds of a spoken language can be visually distinguished and thus enabling deaf individuals to communicate with each other and also with normal-hearing people. Non-Audible Murmur is very quietly uttered speech which can be perceived by a special acoustic sensor (i.e., NAM microphone). NAM microphones can be used for privacy, for robustness against noise, and also by speech-impaired people. In this study, experimental results are also presented showing the effectiveness of the two methods in augmentative speech communication.

2. Cued Speech

To date, visual information is widely used to improve speech perception or automatic speech recognition (lipreading) (Potamianos et al., 2003). With lipreading technique, speech can be understood by interpreting the movements of lips, face and tongue. In spoken languages, a particular facial and lip shape corresponds to a specific sound (phoneme). However, this relationship is not one-to-one and many phonemes share the same facial and lip shape (visemes). It is impossible, therefore to distinguish phonemes using visual information alone. Without knowing the semantic context, one cannot perceive the speech thoroughly even with high lipreading performances. To date, the best lip readers are far away into reaching perfection. On average, only 40 to 60% of the vowels of a given language (American English) are recognized by lipreading (Montgomery & Jackson, 1983), and 32% when relating to low predicted words (Nicholls & Ling, 1982). The best result obtained amongst deaf participants was 43.6% for the average accuracy (Auer & Bernstein, 2007; Bernstein et al., 2007). The main reason for this lies in the ambiguity of the visual pattern. However, as far as the orally educated deaf people are concerned, the act of lipreading remains the main modality of perceiving speech.

To overcome the problems of lipreading and to improve the reading abilities of profoundly deaf children, Cornett (O.Cornett, 1967) developed in 1967 the Cued Speech system to complement the lip information and make all phonemes of a spoken language clearly visible. As many sounds look identical on face/lips (e.g., $/p/$, $/b/$, and $/m/$), using hand information those sounds can be distinguished and thus make possible for deaf people to completely understand a spoken language using visual information only.

Cued Speech [also referred to as Cued Language (Fleetwood & Metzger, 1998)] uses hand shapes placed in different positions near the face along with natural speech lipreading to enhance speech perception from visual input. This is a system where the speaker faces the perceiver and moves his hand in close relation with speech. The hand, held flat and oriented so that the back of the hand faces the perceiver, is a cue that corresponds to a unique phoneme when associated with a particular lip shape. A manual cue in this system contains two components: the hand shape and the hand position relative to the face. Hand shapes distinguish among consonant phonemes whereas hand positions distinguish among vowel phonemes. A hand shape, together with a hand position, cues a syllable. Cued Speech

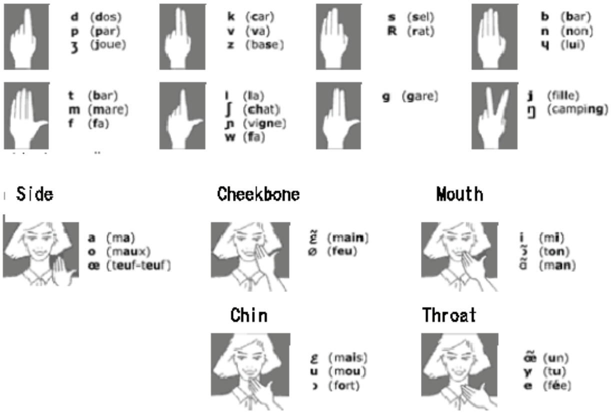


Fig. 1. Hand shapes for consonants (top) and hand position (bottom) for vowels in French Cued Speech.

improves the speech perception of deaf people (Nicholls & Ling, 1982; Uchanski et al., 1994). Moreover, for deaf people who have been exposed to this mode since their youth, it offers a complete representation of the phonological system, and therefore it has a positive impact on the language development (Leybaert, 2000). Figure 1 describes the complete system for French. In French Cued Speech, eight hand shapes in five positions are used. The system was adapted from American English to French in 1977. To date, Cued Speech has been adapted in more than 60 languages.

Another widely used communication method for deaf individuals is the Sign Language (Dreuw et al., 2007; Ong & Ranganath, 2005). Sign Language is a language with its own grammar, syntax and community; however, one must be exposed to native and/or fluent users of Sign Language to acquire it. Since the majority of children who are deaf or hard-of-hearing have hearing parents (90%), these children usually have limited access to appropriate Sign Language models. Cued Speech is a visual representation of a spoken language, and it was developed to help raise the literacy levels of deaf individuals. Cued Speech was not developed to replace Sign Language. In fact, Sign Language will be always a part of deaf community. On the other hand, Cued Speech is an alternative communication method for deaf individuals. By cueing, children who are deaf would have a way to easily acquire the native home language, read and write proficiently, and communicate more easily with hearing family members who cue them.

In the first attempt for vowel recognition in Cued Speech, in (Aboutabit et al., 2007) a method based on separate identification, i.e., indirect decision fusion was used and a 77.6% vowel accuracy was obtained. In this study, however, the proposed method is based on HMMs and uses concatenative feature fusion to integrate the components into a combined one and then perform automatic recognition. Fusion (Adjoudani & Benoît, 1996; Hennecke et al., 1996; Nefian et al., 2002) is the integration of all available single modality streams into a combined one. In this study, lip shape and hand components are combined in order to realize automatic recognition in Cued Speech for French.

3. Non-Audible Murmur (NAM)

Non-Audible Murmur (NAM) refers to a very softly uttered speech received through the body tissue. A special acoustic sensor (i.e., the NAM microphone) is attached behind the talker's ear. This receives very soft sounds that are inaudible to other listeners who are in close proximity to the talker.

The attachment of the NAM microphone to the talker is shown in Figure 2. The first NAM microphone was based on stethoscopes used by medical doctors to examine patients, and was called the stethoscopic microphone (Nakajima et al., 2003). Stethoscopic microphones were used for the automatic recognition of NAM speech (Heracleous et al., 2004). The silicon NAM microphone is a more advanced version of the NAM microphone (Nakajima et al., 2005). The silicon NAM microphone is a highly sensitive microphone wrapped in silicon; silicon is used because its impedance is similar to that of human skin. Silicon NAM microphones have been employed for automatic recognition of NAM speech as well as for NAM-to-speech conversion (Toda & Shikano, 2005). Similar approaches have been introduced for speech enhancement or speech recognition (Jou et al., 2004; Zheng et al., 2003). Further, non-audible speech recognition has also been reported based on electromyographic (EMG) speech recognition, which processes electric signals caused by the articulatory muscles (Walliczek et al., 2006).

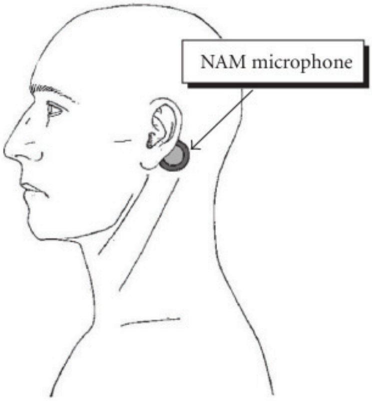


Fig. 2. NAM microphone attached to the talker

The speech received by a NAM microphone has different spectral characteristics in comparison to normal speech. In particular, the NAM speech shows limited high-frequency contents because of body transmission. Frequency components above the 3500-4000 Hz range are not included in NAM speech. The NAM microphone can also be used to receive audible speech directly from the body [Body Transmitted Ordinary Speech (BTOS)]. This enables automatic speech recognition in a conventional way while taking advantage of the robustness of NAM against noise.

Previous studies have reported experiments for NAM speech recognition that produced very promising results. A word accuracy of 93.9% was achieved for a 20k Japanese vocabulary dictation task when a small amount of training data from a single speaker was used (Heracleous et al., 2004). Moreover, experiments were conducted using simulated and real noisy test data with clean training models to investigate the role of the Lombard reflex (Heracleous et al., 2007; Junqua, 1993) in NAM recognition.

In the present study, audio-visual NAM recognition is investigated by using the concatenative feature fusion, the multistream HMM decision fusion, and late fusion to integrate the audio and visual information. A statistical significance test was performed, and audio-visual NAM recognition in a noisy environment was also investigated.

4. Experiments

4.1 Cued Speech automatic recognition

The data for vowel- and consonant recognition experiments were collected from a normal-hearing cuer. The female native French speaker employed for data recording was certified in transliteration speech into Cued Speech in the French language. She regularly cues in schools. The cuer wore a helmet to keep her head in a fixed position and opaque glasses to protect her eyes against glare from the halogen floodlight. The cuer's lips were painted blue, and blue marks were marked on her glasses as reference points. These constraints were applied in recordings in order to control the data and facilitate the extraction of accurate features.

The data were derived from a video recording of the cuer pronouncing and coding in Cued Speech a set of 262 French sentences. The sentences (composed of low predicted multi-syllabic words) were derived from a corpus that was dedicated to Cued Speech synthesis (Gibert et al., 2005). Each sentence was dictated by an experimenter, and was repeated two or three times (to correct the pronunciation errors) by the cuer resulting in a set of 638 sentences.

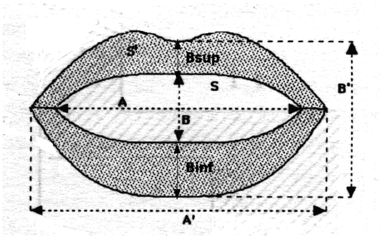


Fig. 3. Parameters used for lip shape modeling.

The audio part of the video recording was synchronized with the image. Figure 3 shows the lip shape parameters used in the study. An automatic image processing method was applied to the video frames in the lip region to extract their inner and outer contours and derive the corresponding characteristic parameters: lip width (A), lip aperture (B), and lip area (S) (i.e., six parameters in all).

The process described here resulted in a set of temporally coherent signals: the 2D hand information, the lip width (A), the lip aperture (B), and the lip area (S) values for both inner and outer contours, and the corresponding acoustic signal. In addition, two supplementary parameters relative to the lip morphology were extracted: the pinching of the upper lip (Bsup) and lower (Binf) lip. As a result, a set of eight parameters in all was extracted for modeling lip shapes. For hand position modeling, the xy coordinates of two landmarks placed on the hand were used (i.e., 4 parameters). For hand shape modeling, the xy coordinates of the landmarks placed on the fingers were used (i.e., 10 parameters). Non-visible landmarks received default coordinates [0,0].

During the recording of Cued Speech material for isolated word recognition experiments, the conditions were different from the ones described earlier. The system was improved by excluding the use of a helmet by the cuer, enabling in this way the head movements during recording. The subject was seated on a chair in a way to avoid large movements in the third direction (i.e. towards the camera). However, the errors that might occur have not been evaluated. In addition, the landmarks placed on the cuer's fingers were of different colors in order to avoid the hand shape coding and the finger identification, and this helped to simplify and speed up the image processing stage. In these recording sessions, a normal-hearing cuer and a deaf cuer were employed. The corpus consisted of 1450 isolated words with each of 50 words repeated 29 times by the cuers.

In the phoneme recognition experiments, context-independent, 3-state, left-to-right, no-skip-phoneme HMMs were used. Each state was modeled with a mixture of 32 Gaussians. In addition to the basic lip and hand parameters, first- (Δ) and second-order derivatives (ΔΔ) were used as well. For training and test, 426 and 212 sentences were used, respectively. The training sentences contained 3838 vowel and 4401 consonant instances, and the test sentences contained 1913 vowel and 2155 consonant instances, respectively. Vowels and consonants were extracted automatically from the data after a forced alignment was performed using the audio signal.

For isolated word recognition experiments two HMM sets were trained (deaf and normal-hearing). Fifteen repetitions of each word were used to train 50, 6-state, whole word HMMs, and 14 repetitions were used for testing. Eight and ten parameters were used for lip shape and hand shape modeling, respectively.

In automatic speech recognition, a diagonal covariance matrix is often used because of the assumption that the parameters are uncorrelated. In lipreading, however, parameters show a strong correlation. In this study, a global Principal Component Analysis (PCA) using all the training data was applied to decorrelate the lip shape parameters and then a diagonal covariance matrix was used. The test data were then projected into the PCA space. All PCA lip shape components were used for HMM training. For training and recognition the HTK3.1 toolkit (Young et al., 2001) was used.

For the integration of the lip shape and hand shape components, feature concatenative fusion was used. Feature concatenation uses the concatenation of the synchronous lip shape and hand features as the joint feature vector

$$O_t^{LH} = [O_t^{(L)^T}, O_t^{(H)^T}]^T \in R^D \tag{1}$$

where O_t^{LH} is the joint lip-hand feature vector, $O_t^{(L)}$ the lip shape feature vector, $O_t^{(H)}$ the hand feature vector, and D the dimensionality of the joint feature vector. In vowel recognition experiments, the dimension of the lip shape stream was 24 (8 basic parameters, 8 Δ, and 8 ΔΔ parameters) and the dimension of the hand position stream was 12 (4 basic parameters, 4 Δ, and 4 ΔΔ parameters). The dimension D of the joint lip-hand position feature vectors was, therefore 36. In consonant recognition experiments, the dimension of the hand shape stream was 30 (10 basic parameters, 10 Δ, and 10 ΔΔ parameters). The dimension D of the joint lip-hand shape feature vectors was, therefore 54. Figure 4 shows the vowel recognition results. As shown, by integrating hand position component with lip shape component, a

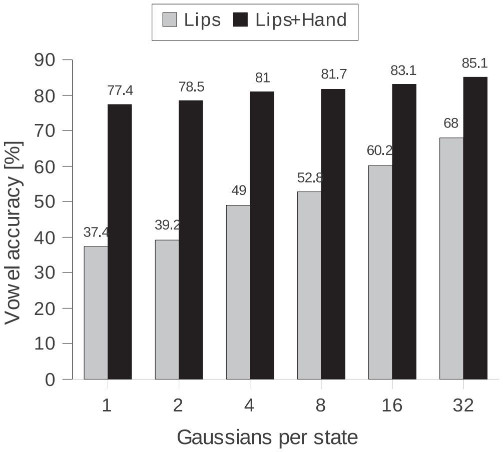


Fig. 4. Cued Speech vowel recognition using only lip and hand parameters based on concatenative feature fusion.

vowel accuracy of 85.1% was achieved, showing a 53% relative improvement compared to the sole use of lip shape parameters.

Using concatenative feature fusion, lip shape component was integrated with hand shape component and consonant recognition was conducted. For hand shape modeling, the *xy* coordinates of the fingers, and first- and second-order derivatives were used. In total, 30 parameters were used for hand shape modeling. For lip shape modeling, 24 parameters were used. Figure 5 shows the obtained results in the function of Gaussians per state. It can be seen that when using 32 Gaussians per state, a consonant accuracy of 78.9% was achieved. Compared to the sole use of lip shape, a 56% relative improvement was obtained.

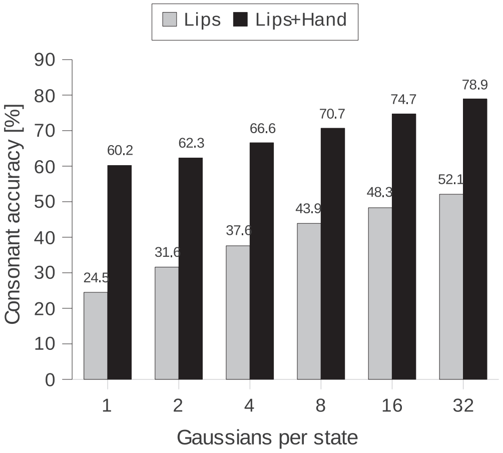


Fig. 5. Cued Speech consonant recognition using only lip and hand parameters based on concatenative feature fusion.

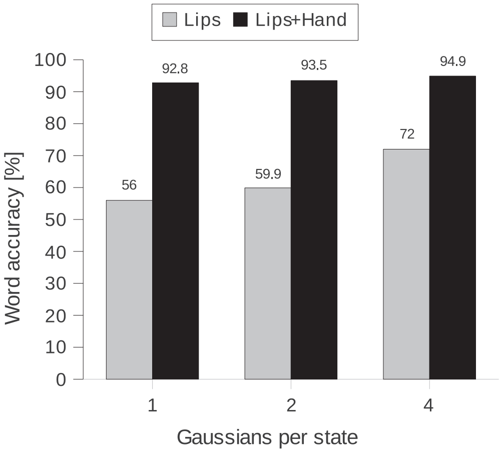


Fig. 6. Word accuracy for isolated word recognition in the case of a normal-hearing subject.

Figure 6 shows the isolated word recognition results obtained in the function of several Gaussians per state in the case of the normal-hearing cuer. In the case of a single Gaussian per state, using lip shape alone obtained a 56% word accuracy; however, when hand shape information was also used, a 92.8% word accuracy was obtained. The highest word accuracy when using lip shape was 72%, obtained in the case of using 4 Gaussians per state. In that case, the Cued Speech word accuracy using also hand information was 94.9%. Figure 7 shows the obtained results in the case of a deaf cuer. The results show that in the case of the deaf subject, words were better recognized when using lip shape alone compared to the normal-hearing subject. The fact that deafs rely on lipreading for speech communication may increase their ability not only for speech perception but also for speech production. The word accuracy in the case of the deaf subject was 89% compared to the 94.9% in the normal-hearing subject.

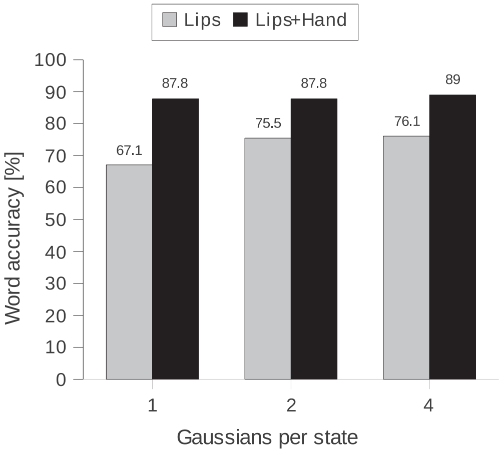


Fig. 7. Word accuracy for isolated word recognition in the case of a deaf subject.

	HMMs		
Test data	Normal	Deaf	Normal+Deaf
Normal	94.9	0.6	92.0
Deaf	2.0	89.0	87.2

Table 1. Word accuracy of a multi-speaker experiment

The difference in performance might be because of the lower hand shape recognition in the deaf subject. It should also be noted that the normal-hearing cuer was a professional teacher of Cued Speech. The results show that there are no additional difficulties in recognizing Cued Speech in deaf subjects, other than those appearing in normal-hearing subjects.

A multi-cuer isolated word recognition experiment was also conducted using the normal-hearing and the deaf cuers' data. The aim of this experiment is to investigate whether it is possible to train speaker-independent HMMs for Cued Speech recognition. The training data consisted of 750 words from the normal-hearing subject, and 750 words from the deaf subject. For testing, 700 words from normal-hearing subject and 700 words from the deaf subject were used, respectively. Each state was modeled with a mixture of 4 Gaussian distributions. For lip shape and hand shape integration, the concatenative feature fusion was used.

Table 1 shows the results obtained when lip shape and hand shape features were used. The results show, that due to the large variability between the two subjects, word accuracy of cross-recognition is extremely low. On the other hand, the word accuracy in normal-hearing subject when using multi-speaker HMMs was 92%, which is comparable with the 94.9% word accuracy when cuer-dependent HMMs were used. In the case of the deaf subject, the word accuracy when using multi-cuer HMMs was 87.2%, which was also comparable with the 89% word accuracy when using speaker-dependent HMMs.

The results obtained indicate that creating speaker-independent HMMs for Cued Speech recognition using a large number of subjects should not face any particular difference, other than those appear in the conventional audio speech recognition. To prove this, however, additional experiments using a large number of subjects are required.

4.2 NAM automatic recognition

The corpus used in the experiment was 212 continuous Japanese utterances, containing 7518 phoneme realisations. A 3-state with no skip HMM topology was used. Forty-three monophones were trained using 5132 phonemes. For the purpose of testing, 2386 phonemes were used. The audio parameter vectors were of length 36 (12 MFCC, 12ΔMFCC, and 12 $\Delta\Delta$MFCC). The HTK3.4 Toolkit was used for training and testing.

The face and profile views of the subject were filmed under conditions of good lighting. The system captured the 3-D positions of 112 colored beads glued on the speaker's face at a sampling rate of 50 Hz (fig. 8), synchronized with the acoustic signal sampled at 16000 Hz. The collection of 30 lip points using a generic 3-D geometric model of the lips is shown in Figure 9 (Revéret & Benoît, 1998).

The shape model is built using the Principal Component Analysis (PCA). Successive applications of PCA are performed on the selected subsets of the data, which generate the main directions. These directions are retained as linear predictors for the whole data set. The mobile points P of the face deviate from their average position B by a linear composition of

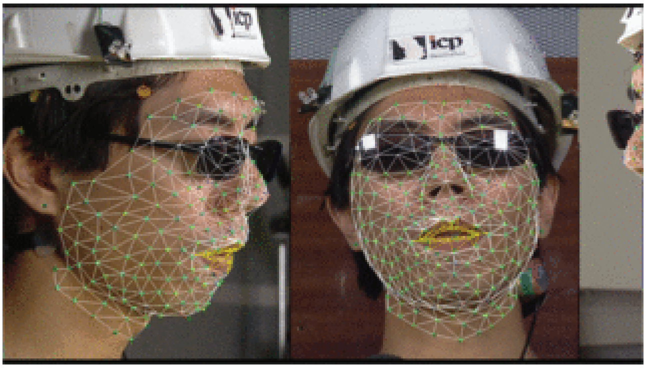


Fig. 8. Characteristic points used for capturing the movements.

the basic components M loaded by factors α (articulatory parameters) (Revéret et al., 2000).

$$P = B + \alpha M \tag{2}$$

Only the first 5 parameters of the extracted 12 linear components M were used. These explained more than 90% of the data variance using the following iterative linear prediction on the data residual: the first component of the PCA on the lower teeth (LT) values leads to the "first jaw" predictor. The PCA on the residual lips values (without jaw1 influence) usually presented three pertinent lip predictors (i.e., lips protrusion, lips closing mainly required for bilabials, and lips raising mainly required for labiodental fricatives). The movements of the throat linked the underlying movements of the larynx and the hyoid bone, and served as the fifth one. The video parameters were interpolated at 200 Hz to synchronize with the audio analysis frame rate. For audio-visual NAM recognition, concatenative feature fusion, multistream decision fusion, and late fusion methods were used.

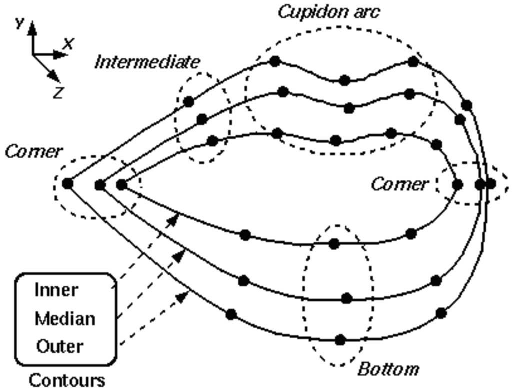


Fig. 9. The 30 control points and the 3 basic contour curves.

Multistream HMM fusion is a state synchronous decision fusion, which captures the reliability of each stream by combining the likelihoods of single-stream HMM classifiers (Potamianos et al., 2003). The emission likelihood of the multistream HMM is the product of the emission likelihoods of the single-stream components, weighted appropriately by stream weights. Given the O combined observation vector, that is, the NAM and visual elements, the emission score of multistream HMM is given by:

$$b_j(O_t) = \prod_{s=1}^{S} [\sum_{m=1}^{M_s} c_{jsm} N(O_{st}; \mu_{jsm}, \Sigma_{jsm})]^{\lambda_s} \tag{3}$$

where, $N(O; \mu, \Sigma)$ is the value in O of a multivariate Gaussian with mean μ and covariance matrix Σ, and S is the number of the streams. For each stream s, M_s Gaussians in a mixture are used, each weighted with c_{jsm}. The contribution of each stream is weighted by λ_s. In the present study, it is assumed that the stream weights do not depend on state j and time t. However, two constraints were applied, namely:

$$0 \leq \{\lambda_n, \lambda_v\} \leq 1, \quad and \quad \lambda_n + \lambda_v = 1 \tag{4}$$

where λ_n is the NAM stream weight, and λ_v is the visual stream weight. In these experiments, the weights were experimentally adjusted to 0.6 and 0.4 values, respectively. The selected weights were obtained by maximizing the accuracy on several experiments.

A disadvantage of the previously described fusion methods is the assumption that there is a synchrony between the two streams. In the present study, late fusion was applied to enable asynchrony between the NAM stream and the visual stream. In the late fusion method, two single HMM-based classifiers were used for the NAM speech and the visual speech, respectively. For each test utterance (i.e., isolated phone), the two classifiers provided an output list, which included all the phone hypotheses with their likelihoods. Subsequently, all the separate mono-modal hypotheses were combined into the bi-modal hypotheses using the weighted likelihoods, as given by:

$$logP_{NV}(h) = \lambda_n logP_N(h|\underline{O}_N) + \lambda_v logP_V(h|\underline{O}_V) \tag{5}$$

where, $logP_{NV}(h)$ is the score of the combined bi-modal hypothesis h, $logP_N(h|\underline{O}_N)$ is the score of the h provided by the NAM classifier, and $logP_V(h|\underline{O}_V)$ is the score of the h provided by the visual classifier. λ_n and λ_v are the stream weights with the same constraints applied in multi-stream HMM fusion.

The procedure described in this study finally resulted in a combined N-best list in which the top hypothesis was selected as the correct bi-modal output. A similar method was also introduced in (Potamianos et al., 2003).

A comparison of the three classification methods used in the present study is shown in Table 2. As seen in the table, the highest classification accuracies are achieved when late fusion is used. The second best classification accuracies are achieved when using multistream HMM decision

	Fusion Method		
	Late	Multistream	Feature
Phonemes	71.8	68.9	67.8
Vowels	86.2	83.7	83.3
Consonants	64.1	59.7	58.2

Table 2. Comparison of the fusion methods in NAM automatic recognition.

fusion. Finally, the lowest accuracies are observed when using feature fusion. Specifically, when using late fusion, an accuracy of 71.8% is achieved for phoneme classification, 86.2% accuracy for vowel classification, and 64.1% accuracy for consonant classification. The highest accuracies, when using late fusion, might be an evidence of asynchrony between the NAM speech and the visual stream. In the following experiments late fusion is used to integrate the NAM audio speech with the visual data. The results obtained when using visual data, NAM data, and visual-NAM data are shown in Figure 10. The results indicate that the classification accuracy is very low when only visual data is used. As many sounds appear to be similar on the lips/face, the sole use of visual parameters cannot distinguish these sounds. In the case of NAM data, the accuracies are higher in comparison to visual data. Specifically, an accuracy of 79.2% was achieved for vowel recognition, 49.8% accuracy for consonant recognition, and 59.7% accuracy for phoneme recognition. It is observed that the accuracy is considerably lower for consonant recognition in comparison to vowel recognition. However, because of the unvoiced nature of NAM, both voiced and unvoiced sounds articulated at the same place become similar, resulting in a larger number of confusions between consonants.The significant improvements in accuracy, when visual data were fused with NAM speech, are shown in Figure 10. Specifically, a relative improvement of 33% was achieved for vowel recognition, 28% for consonant recognition, and 30% for phoneme recognition.

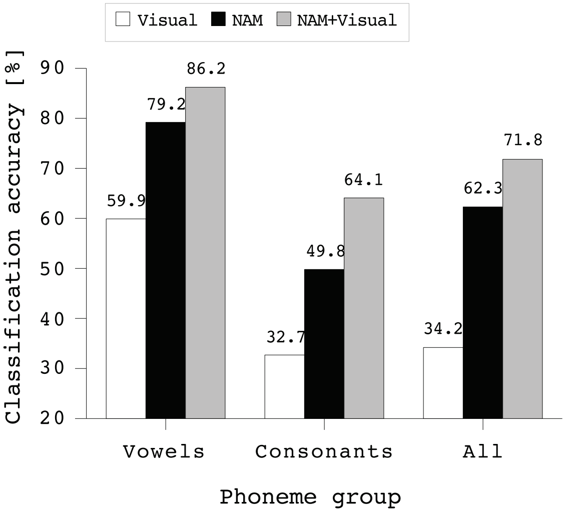


Fig. 10. Phoneme classification in a clean environment.

The McNemar’s test (Gillick & Cox, 1989) was performed to determine whether the differences were statistically significant . The p-values in all the cases were 0.001, which indicated that the differences were statistically significant.

Table 3 and Table 4 show the confusion matrices of the plosives sounds when using NAM and NAM-visual speech, respectively. As is shown, the number of confusions decreases when visual information was also used resulting in a higher accuracy.

In another experiment, office noise recorded by a NAM microphone was superimposed on clean NAM speech on several Signal-to-Noise-Ratio (SNR) levels. The noisy data were used

	/p/	/b/	/t/	/d/	/k/	/g/
/p/	0	0	5	0	2	1
/b/	0	8	5	1	3	0
/t/	2	0	36	3	12	1
/d/	0	2	6	14	4	1
/k/	1	0	8	0	45	6
/g/	0	1	0	2	6	20

Table 3. Confusion matrix of Japanese plosives using NAM speech.

	/p/	/b/	/t/	/d/	/k/	/g/
/p/	3	0	5	0	0	0
/b/	1	13	3	0	0	0
/t/	0	0	39	1	13	1
/d/	0	0	7	17	1	2
/k/	0	0	7	0	50	3
/g/	0	0	0	1	6	22

Table 4. Confusion matrix of Japanese plosives using NAM-visual speech.

to train HMMs of a desired SNR level. In addition, the noisy NAM data were fused with the visual parameters and audiovisual NAM HMMs were trained.

The classification accuracies in the function SNR levels for the visual, the NAM, and the NAM-visual cases are shown in Figure 11 . As seen in the figure, the accuracy of NAM recognition decreases when noisy data is used. However, the accuracy drastically increases when NAM speech is integrated with visual information. In such a case, an average of 15% absolute increase in accuracy was obtained.

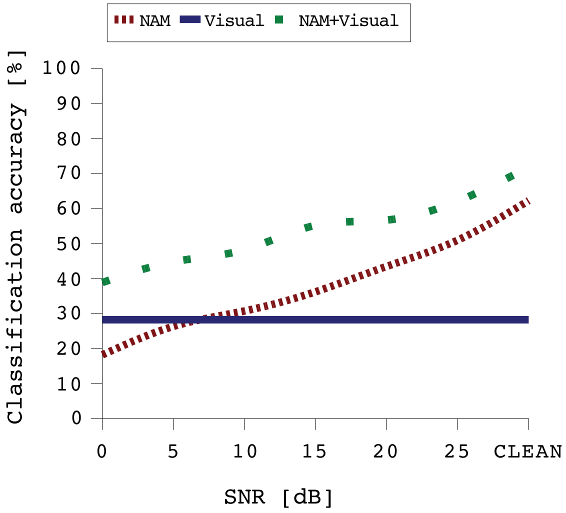


Fig. 11. Phoneme classification in noisy environment.

5. Conclusion and future word

In this chapter, two methods for augmentative speech communication were introduced. Specifically, automatic recognition for Cued Speech for French and Non-Audible Murmur recognition were reported. The authors demonstrated the effectiveness of both methods in alternative speech communication, when modalities other than the audio one are used. Regarding Cued Speech automatic recognition, the experimental results obtained showed recognition rates comparable to those obtained when audio speech is used. In addition, the results showed that using hand information as complement to lip movements, significantly higher rates achieved compared to the sole use of lip movements. With concern to Non-Audible Murmur recognition, the results showed that the unvoiced nature of NAM speech causes a higher number of confusions. Using, however, visual information produced by face/lips further improvements achieved compared with using NAM speech only. As future work, the authors plan to investigate the Cued Speech for Japanese, and also to evaluate the intelligibility of audible NAM speech in clean and noisy environments. This work has been partially supported by JST CREST ’Studies on Cellphone-type Teleoperated Androids Transmitting Human Presence’

6. References

Aboutabit, N., Beautemps, D. & Besacier, L. (2007). Automatic identification of vowels in the cued speech context, *in Proceedings of International Conference on Auditory-Visual Speech Processing (AVSP)* .

Adjoudani, A. & Benoît, C. (1996) On the integration of auditory and visual parameters in an hmm-based asr, *in Speechreading by Humans and Machines, D. G. Stork and M. E. Hennecke, Eds. Berlin, Germany:Springer* p. 461471.

Auer, E. T. & Bernstein, L. E. (2007). Enhanced visual speech perception in individuals with early-onset hearing impairment, *Journal of Speech, Language, and Hearing* 50: 1157–1165.

Bernstein, L., Auer, E. & Jiang, J. (2007). Lipreading, the lexicon, and cued speech, *In C. la Sasso and K. Crain and J. Leybaert (Eds.), Cued Speech and Cued Language for Children who are Deaf or Hard of Hearing, Los Angeles, CA: Plural Inc. Press* .

Dreuw, P., Rybach, D., Deselaers, T., Zahedi, M. & Ney, H. (2007). Speech recognition techniques for a sign language recognition system, *In Proceedings of Interspeech* pp. 2513–2516.

Fleetwood, E. & Metzger, M. (1998). Cued language structure: An analysis of cued american english based on linguistic principles, *Calliope Press, Silver Spring, MD (USA), ISBN 0-9654871-3-X* .

Gibert, G., Bailly, G., Beautemps, D., Elisei, F. & Brun, R. (2005). Analysis and synthesis of the 3d movements of the head, face and hand of a speaker using cued speech, *Journal of Acoustical Society of America* vol. 118(2): 1144–1153.

Gillick, L. & Cox, S. (1989). Some statistical issues in the comparison of speech recognition algorithms, *in Proceedings of ICASSP89* pp. 532–535.

Hennecke, M. E., Stork, D. G. & Prasad, K. V. (1996). Visionary speech: Looking ahead to practical speechreading systems, *in Speechreading by Humans and Machines, D. G. Stork and M. E. Hennecke, Eds. Berlin, Germany: Springer* p. 331349.

Heracleous, P., Abboutabit, N. & Beautemps, D. (2009). Lip shape and hand position fusion for automatic vowel recognition in cued speech for french, *in IEEE Signal Processing Letters* 16: 339–342.

Heracleous, P., Kaino, T., Saruwatari, H. & Shikano, K. (2007). Unvoiced speech recognition using tissue-conductive acoustic sensor, *EURASIP Journal on Advances in Signal Processing* 2007.

Heracleous, P., Nakajima, Y., Lee, A., Saruwatari, H. & Shikano, K. (2004). Non-audible murmur (nam) recognition using a stethoscopic nam microphone, *in Proceedings of Interspeech2004-ICSLP* pp. 1469–1472.

Hueber, T., Chollet, G., Denby, B., Dreyfus, G. & Stone, M. (2008). Phone recognition from ultrasound and optical video sequences for a silent speech interface, *in Proc. of Interspeech* pp. 2032–2035.

Jou, S. C., Schultz, T. & Waibel, A. (2004). Adaptation for soft whisper recognition using a throat microphone, *in Proceedings of Interspeech2004-ICSLP* .

Jou, S., Schultz, T., Walliczek, M., Kraft, F. & Waibel, A. (2006). Towards continuous speech recognition using surface electromyography, *in Proc. of ICSLP* pp. 573–576.

Junqua, J.-C. (1993). The lombard reflex and its role on human listeners and automatic speech recognizers, *J. Acoust. Soc. Am.* 1.

Leybaert, J. (2000). Phonology acquired through the eyes and spelling in deaf children, *Journal of Experimental Child Psychology* 75: 291–318.

Montgomery, A. A. & Jackson, P. L. (1983). Physical characteristics of the lips underlying vowel lipreading performance, *Journal of the Acoustical Society of America* 73 (6): 2134–2144.

Nakajima, Y., Kashioka, H., Shikano, K. & Campbell, N. (2003). Non-audible murmur recognition, *in Proceedings of EUROSPEECH* pp. 2601–2604.

Nakajima, Y., Kashioka, H., Shikano, K. & Campbell, N. (2005). Remodeling of the sensor for non-audible murmur (nam), *in Proceedings of Interspeech2005-EUROSPEECH* pp. 389–392.

Nakamura, K., Toda, T., Nakajima, Y., Saruwatari, H. & Shikano, K. (2008). Evaluation of speaking-aid system with voice conversion for laryngectomees toward its use in practical environments, *in Proc. of Intespeech* pp. 2209–2212.

Nefian, A. V., Liang, L., Pi, X., Xiaoxiang, L., Mao, C. & Murphy, K. (2002). A coupled hmm for audio-visual speech recognition, *in Proceedings of ICASSP 2002* .

Nicholls, G. & Ling, D. (1982). Cued speech and the reception of spoken language, *Journal of Speech and Hearing Research* 25: 262–269.

O.Cornett, R. (1967). Cued speech, *American Annals of the Deaf* 112: 3–13.

Ong, S. & Ranganath, S. (2005). Automatic sign language analysis: A survey and the future beyond lexical meaning, *IEEE Trans. PAMI* vol. 27, no. 6: 873891.

Potamianos, G., Gravier, G., Garg, A., Cooley, A. S. & Tukey, J. W. (2003). Recent advances in the automatic recognition of audiovisual speech, *in Proc. of the IEEE* 91, Issue 9: 1306–1326.

Revéret, L., Bailly, G. & Badin, P. (2000). Mother: a new generation of talking heads providing a flexible articulatory control for video-realistic speech animation, *in Proceedings of ICSLP* pp. 755–758.

Revéret, L. & Benoît, C. (1998). A new 3d lip model for analysis and synthesis of lip motion in speech production, *in Proceedings of AVSP* .

Toda, T. & Shikano, K. (2005). Nam-to-speech conversion with gaussian mixture models, *in Proc. of Interspeech* pp. 1957–1960.

Tran, V. A., Bailly, G., Loevenbruck, H. & Jutten, C. (2008). Improvement to a nam captured whisper-to-speech system, *in Proc. of Interspeech* pp. 1465–1468.

Uchanski, R. M., Delhorne, L. A., Dix, A. K., Braida, L. D., Reedand, C. M. & Durlach, N. I. (1994). Automatic speech recognition to aid the hearing impaired: Prospects for the automatic generation of cued speech, *Journal of Rehabilitation Research and Development* vol. 31(1): 20–41.

Walliczek, M., Kraft, F., Jou, S.-C., Schultz, T. & Waibel, A. (2006). Sub-word unit based non-audible speech recognition using surface electromyography, *in Proceedings of Interspeech2006-ICSLP* pp. 1487–1490.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V. & Woodland, P. (2001). The htk book, *Cambridge University Engineering Department* .

Zheng, Y., Liu, Z., Zhang, Z., Sinclair, M., Droppo, J., Deng, L., Acero, A. & Huang, Z. (2003). Air- and bone-conductive integrated microphones for robust speech detection and enhancement, *in Proceedings of ASRU* pp. 249–253.

16

Soccer Event Retrieval Based on Speech Content: A Vietnamese Case Study

Vu Hai Quan
University of Science, VNU-HCM,
Vietnam

1. Introduction

Video is a self-contained material which carries a large amount of rich information, far richer than text, audio or image. Researches (Amir et al., 2004), (Fleischman & Roy, 2008), (Fujii et al., 2006) have been conducted in the field of video retrieval amongst which content-based retrieval of video events is an emerging research topic. Figure 1 illustrates an ideal content-based video retrieval system which combines spoken words and imagery. Such ideal system would allow retrieval of relevant clips, scenes, and events based on queries which could include textual description, image, audio and/or video samples. Therefore, it involves automatic transcription of speech, multi-modal video and audio indexing, automatic learning of semantic concepts and their representation, advanced query interpretation and matching algorithms, imposing many new challenges to research.

There is no universal definition of video event, and the existing definitions can be classified into two types: one is being abnormal and the other is interesting to users (Babaguchi et al., 2002). In the first type of definition, an event may be either normal or abnormal. Generally speaking, only the abnormal event, which has more information than the normal one, is meaningful to the users. This event definition is suitable for the video analysis under restricted circumstance such as surveillance. The event definition of interesting to users is based on the users' description and domain prior knowledge (Sun & Yang, 2007). Suitable examples of this category are sport-video events such as ones in soccer and baseball. Several popular soccer events are shown in Fingure 2, including scoring, corner kick, yellow card and foul events.

Soccer video analysis plays an important role in both research and commerce. The basic idea of soccer events retrieval is to infer and retrieve the interesting events, and its goal is to make the results accord with human's visual perception as much as possible (Xu et al., 2001). Inference of events can be stemmed from either the semantic visual concepts or the spontaneous speech embedded in the videos. This chapter approaches soccer-video event retrieval in an audio aspect (i.e., the problem of spontaneous speech recognition). In this case, an event is defined as the spatiotemporal entity interesting to users, which is remarked by the announcer's spoken words. By exploiting spoken information of the video, soccer events are detected using an automatic speech recognition (ASR) system. However, as soccer videos vary in both speech quality and content, a canonical speech recognizer would not perform well without modifications and improvements. There are three main problems induced by data diversity: noisy speech, foreign term interferences, and emotional variations in speech prosody.

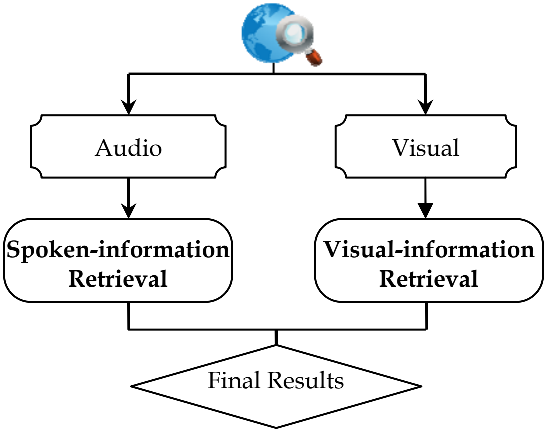


Fig. 1. A full-fledged content-based video retrieval system

To cope with these problems, a noise reduction scheme, a cross-lingual transliteration model, and an advanced acoustic modelling technique are proposed. In the remainder of this chapter, Section 2 gives a detailed specification of the retrieval system. Section 3 focuses on experimental evaluations. And finally, Section 4 concludes the discussions.

(a) Scoring (b) Corner Kick (c) Foul (d) Yellow Card

Fig. 2. Soccer events

2. The retrieval system

This section gives a detailed specification of the proposed retrieval system for soccer video database. Figure 3 illustrates four main parts comprising the system: a speech recognizer, a transliteration model, a noise suppressor, and a search engine. Each one plays an indispensable role in the whole system. The speech recognizer manages video transcriptions with the aid of transliteration model and the noise suppressor, while search engine deals with the tasks of indexing and retrieving transcribed text.

The following subsections will focus on each of the components respectively. As for the search engine, this system makes use of standard information retrieval techniques (e.g., indexing, matching, etc.). Therefore, it will not be covered in this chapter.

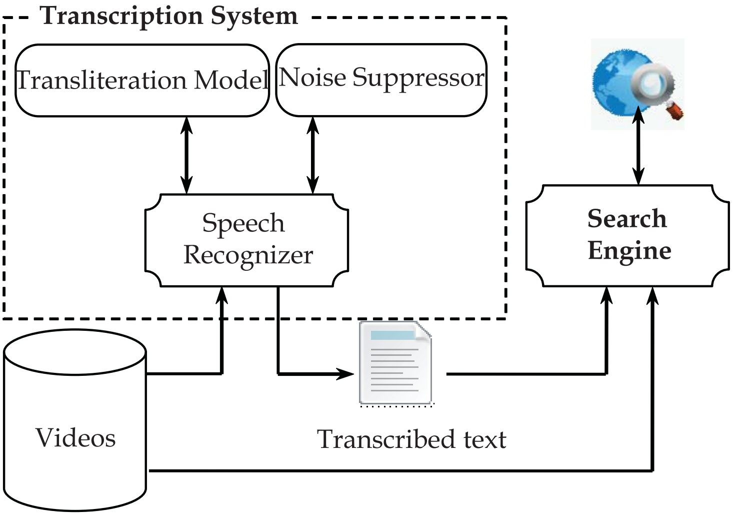


Fig. 3. The retrieval system

2.1 Vietnamese speech recognition

The speech recognition system that was reported in (Vu et al., 2006) is employed as the recognizer for the retrieval system. In this subsection, modifications in acoustic modelling and transcription process to the recognizer are discussed.

2.1.1 Advanced acoustic modeling

The acoustic modelling technique described in (Vu et al., 2006) is designed in the usual approach as for Chinese (Mori et al., 1997) in which each syllable is decomposed into initial (I) and final (F) parts (Figure 4a). While most of Vietnamese syllables consist of an Initial and a Final, some of them only have the Final. The initial part always corresponds to a consonant. The final part includes a main sound plus tone and an optional ending sound. This decomposition has two advantages. First, the number of monophones is relatively small (44 monophones). Second, by treating tone as a distinct phone, followed immediately after the main sound, the context-dependent model for tone can be built straightforwardly. It means that the recognition of tones was fully integrated in the system in just one recognition pass. However, distinct representations of tones have brought upon a disadvantage: the deficiency in modelling emotional variations of speech prosody. Since emotional prosody is expressed in the main tonal sound, separating tone from vowel would degrade the parameterization of tonal vowels.

To better model emotional prosodies, a modification to the acoustic model is proposed, in which tones are integrated into tonal vowels. This results in a new acoustic model consisting of 99 monophones including 27 phones for consonants, 12 phones for non-tonal vowels, and 60 phones for tonal vowels as shown in Figure 4b. Table 1 gives examples showing the differences between tone representations.

S → [I] F

F → V T [E]

I → b c ch d đ g gh gi h k kh l m n ng ngh nh p ph qu r s t tr th v x

V → a ă â e ê i o ô ơ u ư y

T → </> <\> <?> <~> <.>

E → c ch ng nh m n p t

Fig. 4a. Separated tone modeling

S → [I] F

F → V [E]

I → b c ch d đ g gh gi h k kh l m n ng ngh nh p ph qu r s t tr th v x

V → a ă â e ê i o ô ơ u ư y
á ắ ấ é ế í ó ố ớ ú ứ ý
à ằ ầ è ề ì ò ồ ờ ù ừ ỳ
ả ẳ ẩ ẻ ể ỉ ỏ ổ ở ủ ử ỷ
ã ẵ ẫ ẽ ễ ĩ õ ỗ ỡ ũ ữ ỹ
ạ ặ ậ ẹ ệ ị ọ ộ ợ ụ ự ỵ

E → c ch ng nh m n p t

Fig. 4b. Integrated tone modeling

Word	Separated tone	Integrated tone
chào	ch a <\> o	ch à o
chao	ch a o	ch a o
chèo	ch e <\> o	ch è o

Table 1. Examples of tone representations

2.1.2 Video transcription

Speech in soccer videos is different from a typical speech training corpus in terms of quality and speaker-variations. This mismatch leads to serious degradation in system performance. In order to minimize errors, the soccer speech is put through a two-stage recognition process as shown in Figure 5. In the first stage, input speech along with its transcription, produced by the recognizer, are used to modify acoustic parameters. This is indeed the unsupervised mode of acoustic adaptation. Gaussian components of the recognizer are adapted using the maximum likelihood linear regression (MLLR) technique. A global regression class is considered for adapting mean vectors with full transformation matrices. In the second stage, final transcriptions of input speech are generated by the adapted model.

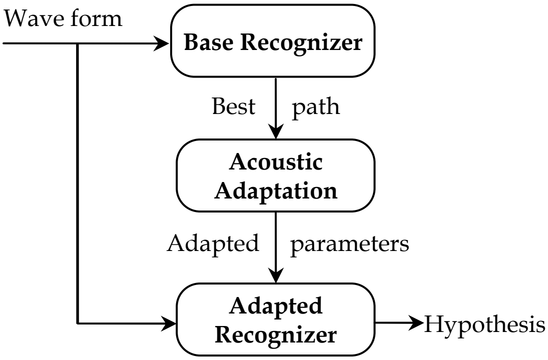


Fig. 5. Two-stage transcription process

2.2 Transliteration of foreign terms

The inability to deal with words in foreign languages causes recognition rates to drop drastically in ASR systems. A common solution to this problem is to look up a pronunciation dictionary. Despite its effectiveness, this approach has serious limitations: making a cross-lingual pronunciation dictionary of large size by hand is costly and required a lot of effort. Furthermore, the number of available entries is finite and therefore not flexible because speech recognition systems are expected to handle arbitrary words. Alternatively, data-driven approaches can be employed to overcome these limitations by learning samples and predicting unseen words. In the retrieval system, joint-sequence model (Bisani & Ney, 2008), a data-driven approach, is applied to transliterate foreign words into Vietnamese syllables.

The fundamental idea of joint-sequence model is based on the concept of graphone, a joint unit between graphemes and phonemes. In the assumption of joint- sequence model, each word and its pronunciation are generated by a common sequence of graphones, but the number of possible graphone sequences varies depending on the ways of segmentation. For instance, the word "David" and its pronunciation can be represented by one of the graphone sequences shown in Figure 6.

Graphone inventory can be estimated from training data using discounted EM algorithm (Bisani & Ney, 2008). The transliteration process searches for the most likely graphone sequence which matches the same spelling as given, and then projects it into phonemes. The resulting phonemes can then be assembled into Vietnamese syllables for speech recognition. It is worth noting that due to the co-segmentation characteristic of graphones, transliteration can be applied bidirectionally. It means that given a sequence of Vietnamese syllables, the corresponding foreign word can be obtained in the same way as presented.

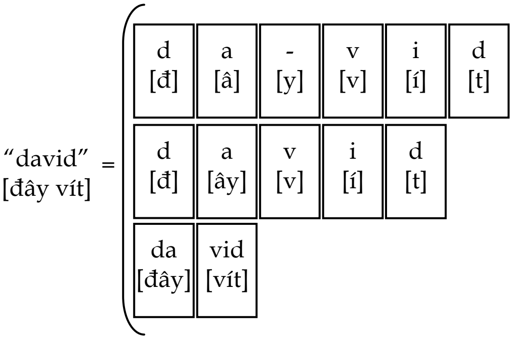

"david" =
[đây vít]
d [đ] | a [â] | - [y] | v [v] | i [í] | d [t]
d [đ] | a [ây] | v [v] | i [í] | d [t]
da [đây] | vid [vít]

Fig. 6. Co-segmentations of the word "David" and its Vietnamese pronunciation

2.3 Noise reduction

Environmental variation has greatly affected the performance of ASR systems. A number of techniques have been proposed for dealing with environmental noise, especially additive noise which commonly plagues sport-domain speech. Additive noise is noise from external sound sources like wind or cheering that is relatively constant and can be modelled as a noise signal that is just added to the clean speech waveform to produce the noisy speech. One of the most popular methods for reducing the effect of additive noise is spectral subtraction (Katagiri et al., 1998). As depicted in Figure 7, the noise spectra S_n estimated during non-speech regions are subtracted from the noisy speech spectra S_y:

$$S_x = S_y - \alpha S_n \tag{1}$$

where α is the scaling factor for emphasis or de-emphasis of the noise spectra. Enhanced speech is then reconstructed based on the resulting spectra S_x.

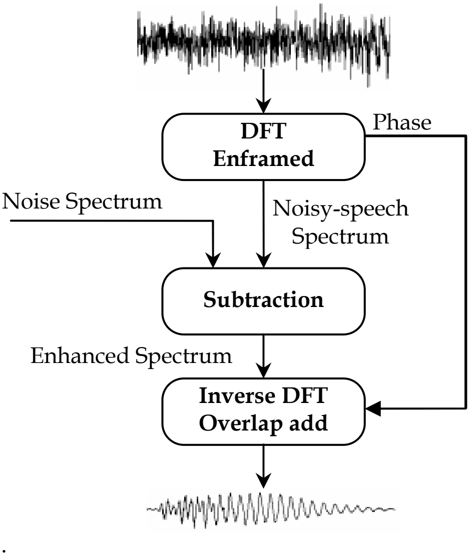


Fig. 7. Spectral subtraction

To minimize the word error rate induced by additive noise contained in soccer videos, magnitude spectrum subtraction is used to enhance speech quality of the videos. In addition, the smoothing technique, that was presented and proved in (Wojcicki et al., 2006) to be effective against the residual effect caused by spectral subtraction, is also employed.

3. Experiments

This section focuses on two main experiments: evaluations of the speech recognizer and the retrieval system. Both of them are conducted on the datasets described below.

3.1 Datasets

3.1.1 Speech and text corpora

The recognizers are trained with the speech corpus that was collected in 2005 from VOV – the national radio broadcaster (mostly in Hanoi and Saigon dialect), with a total duration of 20 hours. It was manually transcribed and segmented into sentences, which resulted in a total of 19496 sentences and a vocabulary size of 3174 words as shown in Table 2. All the speech was sampled at 16 kHz and 16 bits. They were further parameterized into 12 dimensional MFCC, energy, and their delta and acceleration (39 length front-end parameters).

Dialect	Duration	# Sentences
Hanoi	18 hours	17502
Saigon	2 hours	1994
Total	20 hours	19496

Table 2. The VOV speech corpus

Language models (bigram and trigram) for the recognizer are built using the 146M-word text corpus collected from newspaper text sources available on the Internet between 4/2008 – 10/2009. In addition, the text corpus (livescore – 2008) in soccer domain, consisting of 1M words, is also employed for language model adaptation.

3.1.2 Video database

For evaluation purposes, the AFF Suzuki-cup video database (2008) is demuxed into 14-hour speech channels. It is also manually transcribed and segmented into 11593 sentences, with a vocabulary size of 1810 words as shown in Table 3. The speech was sampled at multiple different rates, but was converted to an identical format of 16 kHz and 16 bits. This database will be served as the test-set for every experiment.

Dialect	Duration	# Sentence	# Foreign terms
Mixed	14 hours	11593	892

Table 3. The AFF video database

3.2 Evaluation metrics

Performance of an ASR system is typically measured in terms of word error rate (WER):

$$WER = \frac{S + I + D}{N} \tag{2}$$

where N is the total number of words in the test-set, and S, I, D are the total number of substitutions, insertions, and deletions respectively. This is indeed the edit distance between the automatically generated transcription and the reference one that was manually transcribed. This chapter makes use of word accuracy rate (WAR), which is defined as WAR = 1 – WER, to report performances of the recognizer.

In order to evaluate performances of the retrieval system, event-detection rates are measured in terms of recall and precision which are given by:

$$\text{Precision} = \frac{\#\,\text{correctly retrieved events}}{\#\,\text{retrieved events}} \tag{3}$$

$$\text{Recall} = \frac{\#\,\text{retrieved events}}{\#\,\text{relevant events in the database}} \tag{4}$$

3.3 Transcription evaluation

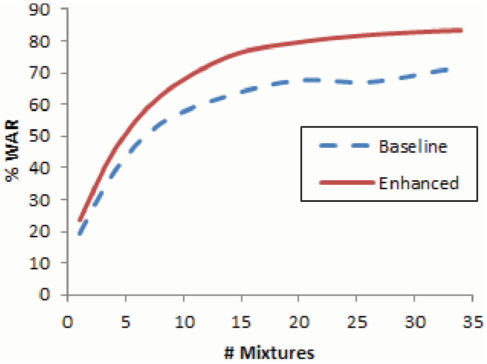


Fig. 8. Performances of the recognizers

In this experiment, the recognizer is evaluated on the task of soccer video transcription. To measure improvements obtained from the proposed methods of transliteration, acoustic modelling and noise reduction, the experiment is conducted in a comparative manner between the original recognizer (without any modifications presented in this chapter) and the modified one. Both are trained using the same corpora described in Subsection 3.1.1. Figure 8 plots performance functions of the two recognizers. As the number of Gaussian mixtures increases, the enhanced recognizer becomes dominant and an improvement of 11.9% can be seen in best case, where WAR reaches 83.3%.

3.4 Event detection evaluation

For evaluations of the retrieval system, Nutch[1] – an open source framework is deployed in role of the search engine. Several typical soccer events are selected as test cases, including: thẻ vàng (yellow card), thẻ đỏ (red card), phạt góc (corner kick), việt vị (offside), phạm lỗi (foul), ghi bàn/bàn thắng (scoring), Công Vinh (a Vietnamese player), Sukha (a Thai player). Table 4 reports their detection rates in the form of recalls along with the corresponding precisions.

[1] http://nutch.apache.org.

Event	# RlE*	# RtE**	# CRtE***	% Recall	% Precision
Yellow Card	32	20	17	62.50	85.00
Red Card	3	2	2	66.67	100.00
Corner Kick	56	38	32	67.86	84.21
Offside	48	34	30	70.83	88.24
Foul	121	65	59	53.72	90.77
Scoring	35	14	12	40.00	85.71
Cong Vinh	153	104	83	67.97	79.81
Su Kha	117	76	57	64.96	75.00
Average	**-**	**-**	**-**	**61.81 ± 9.56**	**86.09 ± 6.96**

*RlE: Relevant events
**RtE: Retrieved events
***CRtE: Correctly retrieved events

Table 4. Event detection rates

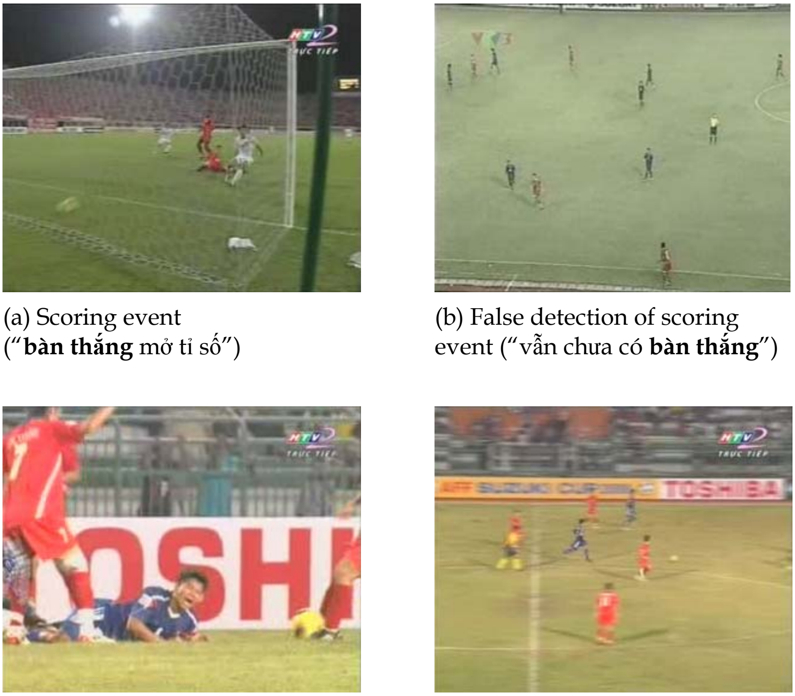
Most of the detection rates (i.e., recall) are above moderate while their precisions are pretty high. The average rates of 61.81% recall and 86.09% precision indicate a reasonable result for the proposed methods and their application in soccer event retrieval. This is indeed the single event detection mode in which each event is defined by a single keyword. Figure 9 gives examples of several single events and their false detections as well. Since events are remarked by the announcers' spoken works, errors in transcriptions will result in missing retrievals. And also, the context in which event-keywords are spoken will be responsible for the false detections. For instance, "scoring/goal" could be spoken in a regular comment (e.g., "vẫn chưa có bàn thắng/still no goal") rather than an authenticated scoring event.

Another way of retrieving soccer events is to combine several keywords together. These events will be denoted as "combined events." Figure 10 illustrates several combined events along with their false detections. Most of the false detections are caused by unexpected combinations between keywords in the results. For example, the combined query "Cong Vinh" & "yellow card" can be resulted in "a yellow card for player A for an unfair act with Cong Vinh" rather than the expected event "a yellow card for Cong Vinh." Someone may suggest enforcing phrase querying, but then again the phrase might not match the announcers' spoken phrase.

Event	# Retrieved events	# Correctly retrieved events	% Precision
"Cong Vinh" & "Yellow Card"	4	2	50.00
"Cong Vinh" & "Offside"	7	7	100.00
"Cong Vinh" & "Foul"	9	7	77.78
"Cong Vinh" & "Scoring"	10	2	20.00
Average	**-**	**-**	**61.95 ± 30.01**

Table 5. Performance of combined-event retrieval

Table 5 summarizes the performances of combined-event retrieval. Since the total number of retrieved events might exceed the number of relevant events, only precisions are reported.



(a) Scoring event
(“**bàn thắng** mở tỉ số”)

(b) False detection of scoring event (“vẫn chưa có **bàn thắng**”)

(c) Foul event
(“**phạm lỗi**”)

(d) False detection of foul event
(“không có **phạm lỗi**”)

Fig. 9. False detections of single events.

3.5 Running-time evaluation

In this experiment, the retrieval system is evaluated in the manner of searching speed. Test cases/queries are generated randomly with respect to both single and combined events. Arbitrary phrase queries (with average syllable length of five) are also taken into account. Table 6 reports the average searching time for single keywords, combined keywords, and arbitrary phrases each with 200 different queries. All the tests were conducted in a standard server with a 16x3.02GHz processor and 32GB RAM.

Category	**Searching time (seconds)**
Single keyword	0.15
Combined keyword	0.24
Arbitrary phrase	0.31
Average	**0.23 ± 0.07**

Table 6. Average searching time

(a) "Cong Vinh" & "scoring" event ("**bàn thắng** của **Công Vinh**")

(b) False detection of "Cong Vinh" & "scoring" event ("chút xíu nữa là **Công Vinh** đã có **bàn thắng**")

(c) "Cong Vinh" & "yellow card" event ("một chiếc **thẻ vàng** cho **Công Vinh**")

(d) False detection of "Cong Vinh" & "yellow card" event ("với pha phạm lỗi với **Công Vinh** trước đó thì Alam Shah đã phải nhận **thẻ vàng**")

Fig. 10. False detections of combined events

A demo version of this system is available for testing at:
www.ailab.hcmus.edu.vn

4. Conclusion

This chapter has presented a spoken information based approach for the retrieval of soccer video events – the first one to apply ASR in sport event retrieval. The entire retrieval system is centred on an automatic speech recognizer. To be applicable in the soccer domain, three modifications for the recognizer are proposed to resolve the problems of noisy speech, foreign term interferences, and prosody variations. Experiments on the video database give reasonable results for the proposed methods. In the near future, this system will be incorporated with the visual-information retrieval system to provide a flexible mechanism for the detection of semantic video events.

5. Acknowledgments

This work is part of the national key project no.KC01.16/06-10, supported by the Ministry of Science and Technology.

6. References

Amir, A., et al. 2004. A multi-modal system for the retrieval of semantic video events. *Computer Vision and Image Understanding*. 96, 2 (Nov. 2004), 216-236.

Babaguchi, N., Kawai, Y. and Kitahashi, T. 2002. Event based indexing of broadcasted sports video by intermodal collaboration. *IEEE Transactions on Multimedia*. 4, 1 (2002), 68-75.

Bisani, M., Ney, H. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*. 50, 5 (May 2008), 434-451.

Fleischman, M., Roy, D. 2008. Grounded language modeling for automatic speech recognition of sports video. In *Proceedings of ACL-08: HLT* (Columbus, OH, 2008). 121-129.

Fujii, A., Itou, K., and Ishikawa, T. 2006. LODEM: A system for on-demand video lectures. *Speech Communication*. 48, 5 (May 2006), 516-531.

Katagiri, E. S., Wan, E. A., and Nelson, A. T. 1998. Networks for speech enhancement. *Handbook of Neural Networks for Speech Processing* (1998).

Le, T., Nguyen, H., and Vu, Q. 2006. Progress in transcription of Vietnamese broadcast news. In *Proceedings of the International Conference on Communications and Electronics* (Hanoi, Vietnam, October 10 - 11, 2006). 300-304.

Mori, R. D., et al. 1997. *Spoken Dialogues with Computers*. Academic Press, San Diego, CA, USA.

Sun, X. H., Yang, J. Y. 2007. Inference and retrieval of soccer event. *Communication and Computer*. 4, 3 (Mar. 2007), 18-32.

Wojcicki, K. K., Shannon, B. J., and Paliwal, K. K. 2006. Spectral subtraction with variance reduced noise spectrum estimates. In *Proceedings of the 11th Australian International Conference on Speech Science & Technology* (Auckland, New Zealand, December 06 – 08, 2006). 76-81.

Xu, P., Xie, L. X., Chang, S. F., Divkaran, A., Vetro, A., and Sun, H. 2001. Algorithms and system for segmentation and structure analysis in soccer video. In *Proceedings of IEEE International Conference on Multimedia and Expo* (Tokyo, Japan, 2001). 576-579

17

Voice Interfaces in Art – an Experimentation with Web Open Standards as a Model to Increase Web Accessibility and Digital Inclusion

Martha Gabriel
University of São Paulo,
Brazil

1. Introduction

The web has been largely mute and deaf but since the beginning of the 21st century this scenario is changing with the possibility of using intelligent voice interfaces on web systems. In this paper we present the *Voice Mosaic* – a system that allows voice interactions on the web through the telephone. Its voice interface uses speech recognition and synthesis solutions developed with VoiceXML, an open-standard in voice technologies adopted by the W3C. *Voice Mosaic* is an artwork that allows people to get in touch with the possibility of talking to the web, intending to cause awareness about it. Since the technology used in *Voice Mosaic* can be used to improve accessibility (for visual impaired people) and digital inclusion (since the telephone is one of the cheapest devices in the world), dissolving borders and amplifying the pervasiveness, we believe that the concepts presented here can be useful to other developers.

2. Voice Interfaces

Voice interfaces are a fascinating subject. The human dream of talking to computers in a natural way is not new. Science fiction books and movies that live in our imagination present several examples of this aspiration, as old television and movie series like "Star Trek," where the Enterprise's staff talk to the ship systems and androids like commander DATA; "Lost in Space," where Will Robinson had in his robot a very loyal and confident friend; the conversations and human interactions with the robots C3PO and R2-D2 in "Star Wars"; "Blade Runner" and its androids and voice driven interfaces; among others (Perkowitz, 2004).

Until recently, talking to computers was in the realm of fiction – the web has been largely mute and deaf. However in the beginning of the 21st century talking to computers has become possible and easy due the enormous advances in speech synthesis and voice recognition technologies as well as the open standards adopted by the W3C (such as VoiceXML). The accuracy level reached by voice technologies now has allowed us to use them widely on the web.

The potential of using voice interfaces is explosive. From speech-only applications integrated to the whole web, to multi-modal applications combining aural and visual abilities into web browsers, voice interfaces add to the flavor of the web a fundamental spice, which is surely going to impact it.

Tim Berners-Lee said at SpeechTEK 2004, NY- "Speech technology is an important ingredient for the Web to realize its full potential." In fact, voice interfaces on the web bring undeniable resources for several areas, as convenience for mobile users, v-commerce, natural interactions, and usability. Beyond the more obvious utilizations for voice interfaces, the ability to talk to the web also provides an important way to improve web-accessibility – not only by multi-modal applications, but also through speech-only ones. Besides that, speech-only applications liberate users from any client computer device to access the internet – in this case, all they need is any telephone in any place in the world. In this sense, since the telephone is one of the cheapest devices in the world, voice interfaces can help improving digital inclusion. This is the alliance of the widest computing network with the most pervasive communication device on Earth – internet & telephone.

However, talking to computers adds "ears" and "mouths" to the Internet organism, changing the way we interact with it, bringing new possibilities and new challenges as well. We must face the increasing complexity that voice interfaces bring to the web while we also open new channels for digital inclusion, provide more accessibility and increase mobility through voice. All these things affect the human role inside the high-tech social structure we live in, at once causing excitement and fear.

2.1 Voice interfaces characteristics

Voice interfaces are specialized computational systems that allow that dialogs happen between human beings and computers (other computational systems) in a way that the computational commands be synthesized in voice in order to be understood by humans, and the human speeches be recognized and transformed into computational codes by the computers. In this way, for instance, instead of accessing a visual page on the web via a browser to fill in a form to book a flight, one could do it via a voice interface, talking to the page.

Voice interfaces are exclusive in a way since they are based on the spoken language. The oral communication plays a big role in the human daily life. Since the youngest ages we spend a substantial part of our waken hours in conversations (Cohen 2004: 7).

According to Pinker (2002: 10), language is an instinct – a biological adaptation to convey information. The idea of the language as a kind of instinct was mentioned for the first time by Darwin in 1871 and in the 20th century, the most famous thesis about the language as an instinct was created by Noam Chomsky (Pinked, 2002: 14). Being the language a natural instinct, it is no wonder that since the machines exists in the human imagination, the utilization of the natural language to talk to them is a latent desire. According to Wilson:

> "The ability of producing and understanding the spoken language was identified by anthropologists as one of the main realizations of our species. Other animals, like dolphins and the primates, may have significant capabilities of vocal communications, but they not get close to the human capabilities." (Wilson, 2002: 775)

However, in despite of the fact that to speak is the most natural and human way of interacting, to access the internet via voice interfaces is as different from navigating on the web via a visual browser as to talk on the telephone is different from reading a letter.

Nowadays we are used to 'browse' and 'write' on the web, which is very different from 'talking' to the web.

Thinking about the differences from visual and aural interfaces, we can start listing the particularities of the voice characteristics. The first one is the transiency. As soon as it is spoken or listened to, the voice disappears and demands that we remember what has been said. On the other hand, visual elements are persistent:

> "The voice is an one-dimensional media with zero persistence. The computational monitor is a bi-dimensional media that combines persistence (you can look to them as long as you want) with selective actualization (you can type a value in any field of the screen without changing the rest of it." (Nielsen, 2003).

Adding to that, according to Santaella:

> "The first principle of sonority is in its evanescence, something that the passage of time makes disappear (…), and the first principle of visuality is in the form that makes itself present before our eyes." (Santaella, 2001: 369).

The second particularity of the voice is the invisibility. That makes it more difficult to indicate to the user which options he can execute and what he needs to say in order to execute them. In visual interfaces, we can always have visible menus and instructions that follow each step of the process in execution.

The third particularity is the asymmetry of the voice. The voice can be produced much faster than being understood; an user can speak faster than typing; and an user can listen slower than reading. In visual interfaces, the user has his own interaction pace to synchronize before continuing the process. In voice interfaces, the pace is not always controlled by the user, but by the interface.

We could say that, according to Cohen (2004: 6), there are two possible modalities of voice interfaces regarding the human senses used – 1) purely aural: where all the process occurs only via sounds and the orality of the speech, without using any visual support, and; 2) multimodal: where the process of interfacing via voice is supported by some kind of visual system associated to it. In the first case, pure voice interfaces, we can mention as an example the access to a system via telephone (see, for example, the artwork Voice Mosaic ahead in this chapter). In the second case, we can mention as an example the multimodal browsers like Opera, which allows access to visual information while one interacts simultaneously via voice (see the application *Multimodal Chinese Food*, using the Opera browser at [http://www-306.ibm.com/software/pervasive/multimodal/chinese/], developed by IBM).

The methodologies and principles for voice interface (VUI – Voice User Interface) design overlap substantially with other types of interface design. However, there is an amount of characteristics of voice interfaces that presents unique design challenges and opportunities. Two of these characteristics stand out when the modality of the interface is purely aural and the interaction happens via spoken language (Cohen, 2004: 6).

Besides the fact that the particularities of the voice affect the purely aural voice interfaces, their operation also differs from the visual interfaces, according the table 1.

According to the Gartner Group (Farber, 2004), in 2015 the interfaces will be invisible and ubiquitous. Although the sensors are the main responsible for the transparent interface of the future, probably the voice interfaces will have their share of responsibility in this process too, since the invisibility is one of the aural characteristics of the pure voice interfaces.

	Visual Interfaces	**Pure Aural Voice Interfaces**
Based on	Visual pages	Blocks of dialogs
Designed for	Control by the eyes	Control by the ears
User action	Brain/ touch (mouse clicks / typing)	Brain / speech
User control	Multi-task (several windows /screens simultaneously)	Mono-task (one conversation a time)
Interaction control	User (the user controls the visual browser – user in command)	Computer (the server controls the voice browser – the browser controls the process)

Table 1. Comparison between the functioning of visual and pure aural voice interfaces

Kerckhove (2003: 21) says that "apparently in the western art and history during the ancient times and, again, from the Renaissance to the modern times, the dominant sensorial bias has been the vision. (…) Nowadays, thanks to the electricity, the actual dominant bias is challenged by the tactile bias" (2003: 21), since we use mouse, keyboard, etc., during most of our computational interaction processes. If we think about the invisible interface we should remember that invisible is not the same as inexistent. Invisible can be immaterial, but the possibility of projecting visual interfaces on *eyesphones*[1], for example, combines the trend of sensors and invisible computers with the human dominance visual and tactile.

Johnson (2001: 101) argues that "simple words keep playing an enormous role in the interface nowadays. And this role seems fated to become more decisive to our informational space in the next decade." Considering that the text editor affects profoundly our way of creating and writing, and that each modality of interface changes our way of thinking and acting in the world, it is expected that all kind of interfaces co-exist and bring hybridizations of the media and forms, as it happened with the email, which, due its frailty and digital form, created a more casual and colloquial style of writing, a mix of the written letter with the talk on the telephone (Johnson, 2001: 105).

In our actual technological scenario, Wilson states that:

> "Computers have a conceptual background from its historical origins from commercial companies and military. The computer screen and its conventions derive from the long history of the representation in the Western culture, from painting, perspective, photography, cinema, to graphic animations and computer metaphors. Similarly, the computer conventional physical interface with keyboard and mouse has a significant cultural baggage. Its restrictions have limited the imagination in thinking about ways of integrating the digital information to human life. (…) Researchers and artists have started to question how the interface between digital systems and people could extend more widely in the human life. Going beyond keyboard and mouse, how the computers could read the human actions such as movement, gesture, touch, look, speech and interactions with physical objects? The wearable computer can convert the body action into information function." (Wilson: 2002: 729).

[1] *Eyesphones* are small glasses that can be connected to computers that project the screen in front of the eyes.

Voice interfaces are a new option in the actual scenario. According to Wilson (2002: 775), "the extension of speech to machines will mark a significant cultural event that will mobilize the artistic attention."

Considering that to "speak" is not the same as "reading" and "writing", and that these processes co-exist along the human history since the most ancient known references, we could suggest here that the most likely scenario is that all different kinds of interface – visual, oral, sensorial, tactile, gestural, etc. – co-exist in the future to answer to the different human needs, instead of replacing each other. Of course, each new kind of interface brings some benefits that answer to more specific needs, but the human needs are diverse and varies according to the context, culture and convenience.

According to Nielsen (2003), "Voice interface will not replace the screens (visual interfaces) as a matter of choice of most users. (…) Several people have overrated impression about rhe benefits of voice interfaces, probably based on the prominence of the voice operated computers in Star Trek." Nielsen also points out that voice interfaces have their great potential in the following cases:

- Users with disabilities that do not allow them to use mouse and/or keyboard or that cannot see;
- Users in situations with busy eyes or hands, for example when driving a car or fixing a complex equipment;
- Users that do not have access to a keyboard and/or monitor, and therefore could use a telephone.

Therefore, for general applications, we believe that voice interfaces can be a great promise as an additional component to multimodal dialogs, more than as an unique interface channel. However, in the case of users with visual or manual disabilities, voice interfaces can be an important channel for inclusion and accessibility.

A research conducted at University of Mariland, consisting in a functional experience for comparing voice controlled web browsers (in the multimodal mode) with mouse controlled web browsers, showed that the voice control improved the performance time in approximately 50% for some kinds of tasks. Subjective measures of satisfaction indicate that for voice navigation, text links are preferable to numeric links, but yet the mouse navigation is still easier to use for general purpose navigation on the web (Christian, 2000).

We can highlight other possibilities for voice interfaces, such as in situations where users prefer to talk to the computer instead of talking to people, as mentioned by Cohen (2004: 9): when the subject of the conversation can cause some kind of embarrassment to the user (for example, when he wants to know about financing values for longer periods and get uncomfortable to ask about low financial rates or about too many options), the user prefer to talk to a voice interface. Although this factor is not exclusive related to voice interfaces, being present in any impersonal man-computer interface, the fact of being able to use natural language to "talk" to a computer about the embarrassing subjects as if one was talking to another person, can provide a better experience that is attractive and pleasant and at the same time answers to the user needs (Cohen, 2004: 11).

According to Nass & Brave (2005), people are 'activated by voice': we respond to voice technologies as we respond to people and we behave as we were in any social situation, and, therefore, the voice interfaces can really emerge as the next frontier for a efficient and friendly technology.

Considering that telephones – either fixed lines as cell phones – exist in larger number and with more penetration in the planet than computers (some places in Africa, for example, where there are not computers available, have a big probability of having telephones available), we can say that voice interfaces reach wider than visual interfaces.

Thinking about the artistic possibilities that the voice interfaces bring, beyond of talking to computers in natural language itself, we could use several voice characteristics regarding the production of artworks, as aesthetical and informational potential. It has become possible in the speech synthesis the manipulation of tone, gender, volume, speed, intonation and voice stress and it can be used to create different perceptions and reactions in the interactors. This possibility of manipulation of vocal characteristics allows generating a dynamic narrative (even in real-time) for stories allowing the use of different personages, according to different contexts and situations. Besides that, in a given moment, we could use phrasal loopings, in another moment, phrases that overlap with each other creating a tridimensional space – louder in the foreground, softer in the background, associating with other temporalities. The sounds could be used accompanied with visual elements in a way that the tridimensional environment would get sound spatiality, and so on.

In the voice interfaces, and probably in any interface, 'what is said' (content) is the most important question in the interaction functional project, and the most important factor that determines usability, according Nielsen (2003). Therefore, the voice interface do not liberate us from the most substantial problems related to interface design: 1) to select the tasks to be supported; 2) to determine the structure of the dialog; 3) to decide which commands or functionalities will be available; 4) to let the users specify what they want, and; 5) to make that the computer give feedback about its actions. As previously mentioned, according to Cohen, "The methodologies and principles of voice interface design substantially overlap with those used in other kinds of interface design. However, there are a number of characteristics in voice interfaces that pose unique challenges and opportunities" (2004:6). Although the main focus of this text is not the interface project itself, it is important to highlight here, as mentioned in Cohem (2004: 4) that the understanding of basic human capabilities and the user needs and goals are the keys for a successful interface design.

The introduction of intelligent voice technologies in the present scenario increases the sonority complexity when compared with previous computational stages, because besides working like an instrument that allows the extension of hearing capabilities, they also allow the complete digitalization and inscription of the voice into computational language, together and mixed with the verbal language (commands or voice information recognized by intelligent voice interfaces become commands or verbal or textual data). According to Santaella (2001: 371), "The verbal language is the most mixed of all languages, because it absorbs the syntax of the sound domain and the form of the visual domain."

Voice interfaces are a new step and possibility for the human-computer interaction, in a process of dissolving the border line between telephone and the internet, and co-existing with other types of interfaces. It is clear that they find their biggest potential in activities and applications in which the modality is auditory and the interaction happens through the spoken language. Due its own peculiar characteristics, the voice brings new artistic potentialities associated with other limits, specialties and complexities, and allow, through the convergence that the technology permits, a new way and new media for communicating, interacting and creating.

2.2 Hypermedia

It is well known that the internet is formed by several servers and clients, and that the most common types of clients, so far, are based on visual interfaces, like email clients (for example, Outlook Express or Gmail), web browsers (for example, Internet Explorer, Firefox, etc.), telnet clients (as Hyperterminal), etc. On the other hand, voice interfaces add one more kind of client to the network, not affecting its topology in terms of servers, but changing drastically the client.

The voice client, i.e., the voice browser, can be a hardware (like the telephone), a software that emulates the telephone (like VoIP softwares, for example), a multimodal browser (like Opera, for example), or even computational devices (like microphones/speakers). Although in voice interfaces the clients are different from those that use visual interfaces, we can have the same applications using simultaneously these two kinds of interfaces. According to Palazzo (2002), the hypertextual system architecture is divided in three levels: presentation level, abstract machine level and database level. Voice interfaces are in the presentation level and can change the system structure specifically in that level, leaving the other levels intact as shown in the figure 1.

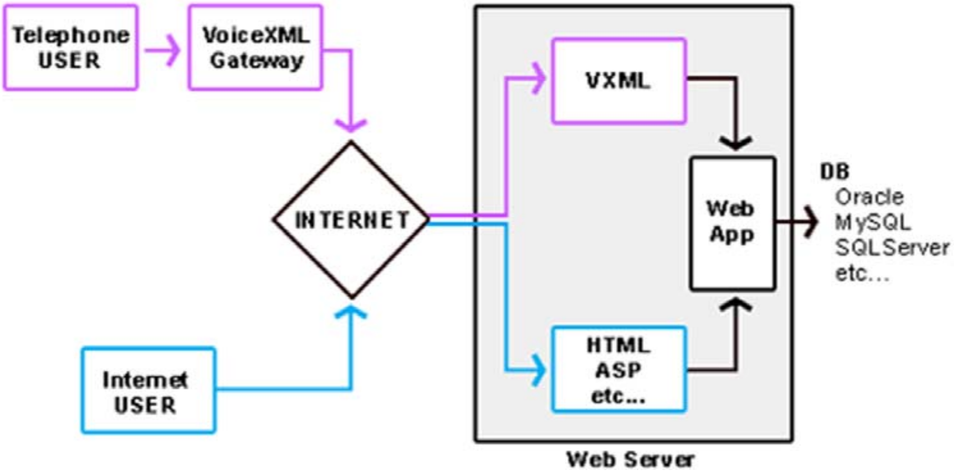


Fig. 1. Diagram showing how voice interfaces created with VoiceXML works in the presentation level, simultaneously with the visual browser.

As can be seen in the figure 1, the line "Telephone USER / VoiceXML Gateway / VXML" represents the application presentation level when accessed by the voice interface, via telephone. The line "Internet USER / HTML" represents the application presentation level when accessed by a visual web browser. The abstract machine level is the box "Web App", which is accessed in the same way by both kinds of interfaces – visual and voice interfaces. From the abstract machine happens the accesses to the databases – the database level – that can also be the same for any kind of interfaces, regardless which technologies are used on the three levels – presentation, abstract machine and database.

When the system (web app) accessed by different kinds of interfaces (visual and aural) is the same, the data captured in the interfaces cannot differ neither in quality nor in quantity, since they are the necessary input for the abstract machine. However, the data input usually needs completely different treatment between visual and aural interfaces. For example, in a visual form it is possible to present all European countries at once as options to be selected by the user. To create a way for the user to choose an European country in a voice interface, maybe the options must be divided in regions (like North, South, East, West), and then be refined to allow the choice of the country. Once the need data (for example, the choice of the European country chosen by the user) is available, regardless where it came from (visual or aural interface), it goes to the same abstract machine. In those cases, the same application can be used simultaneously by both interfaces – visual and aural.

The fact that we can have multiple interfaces without having to change the abstract machine is a very important factor to allow systems integration and hybridization, since it is necessary that there be some common level in the intermixed systems in order to allow the intermixing process.

Of course, voice interfaces allow specific functionalities and data inputs that are not possible in visual interfaces, like voice recording, for example. Applications that benefits from voice interfaces specific capabilities may eventually need to capture / process / store data in specific ways too, separately from the main application, in order to be able to deal with those specific data. However, this situation of having specific data to treat is not new – even in visual systems, according to their devices and its technological capabilities, it is very common to find diverse characteristics in diverse devices/interfaces that need a different treatment in parallel with the main system. One example of that are the web browsers for smartphones that many times have several limitations and/or differences regarding the type of information they can provide to the main systems, when compared with desktop computational browsers. However, the challenge is to keep the same abstract machine and database levels regardless the kind of interface or device in the presentation level, even if it means to build broader databases that comprise optional specific data according to the interface type. That trend, regulated by the W3C (World Wide Web Consortium) open standards, aims to allow a bigger web integration and interoperability, making it easier the convergence and hybridization.

Voice interfaces, therefore, work like an entrance door or like an initial center for the system that connects the user to a larger hypermidiatic context on the web, in a way that the interactor can 'write' his path on the web in a non-linear way starting from the available options in the voice interface. However, when analyzing the 'reading' process starting from the decisions taken in the voice interface, there is no way to escape from the linearity inherent to orality: the options are presented in a linear order to allow a subsequent non-linear choice by the user.

Considering that the orality presents linear characteristics regarding the user reception, one could initially conclude that voice interfaces would not be hypermidiatic systems and that the complexity level would be small when compared to visual interfaces in the network. However, linearity is just a particular case of non-linearity and voice interfaces are part of bigger hypertextual systems, acting as nodes and transitory centers, connecting the interactor to the further levels in the network. According to Murray (2003:10), the term 'non-linear' should be replaces by 'multisequencial' and 'multiform', as expressions to understand the new narratives forms that: allows the ability to navigate through inter-crossed paths from different points of view, in the first case; and in multiple versions generated from the same fundamental representation, in the second. Besides that, the voice particularities – transience, invisibility, asymmetry, imperfections and limitations – increase the complexity level of the system and, consequently, the necessity of organizational rigor. The voice asymmetry, for example, requires a rigorous voice interfaces analysis and program in order to adequate the rhythm of voice reproduction/comprehension.

Although the presentation of each level of a voice interface is done through the linear orality, the user navigation between levels follow a hypertextual path, i.e., non-linear, which can even intercross and interconnect with visual and/or hybrid systems in the network, making the complexity even bigger. An example of such hybrid system is the artwork Voice Mosaic (that will be presented ahead in this text), in which the options and fragments of recorded voices made in the system through a voice interface accessed by phone, configure and present visual and aural records in a visual web interface. The signs – visual and aural – stored in the same database are accessed, generated and experienced by two distinct hypermidiatic interfaces – the voice and the visual ones.

Due the voice transience, the paths and options presented during the speech in a voice interface need to be kept in our short memory. In order to be accessed in a comfortable way to be used, it is necessary that the amount of those paths and options be much more limited than in a visual interface, where they can be presented on a computer screen not needing to be transferred to our short memory. As studied by Miller (1956) and explained by Zakia (1997: 82), we can see that we are limited regarding the amount of information we can keep correctly in our short memory:

> "All of our senses are connected in memory. We have memories not only for visual experiences but also of experiences involving sound, smell, taste, touch, movement, and balance. The memories we remember for a long time are called long-term memories (LTM) and are contrasted with short-term memories (STM) that we remember just long enough to use and then forget. (…) There is a limit to the amount of unrelated information a person can hold in STM, from five to nine items, averaging seven."

Therefore, our capability for using spoken options is smaller than our capabilities for using them in the visual mode, where besides of not requiring that the options be all memorized (since they are persistent in the visual and not transient like the voice), it also has a larger associated memory to it.

2.3 Interactivity

Although voice interfaces provide a more natural and human way of interaction between man/computer, they present several differences from visual interfaces.

The first difference is related to how visual screen browsers (like Firefox or internet Explorer, for example) and voice browsers (like the telephone) works. In the first case, the user has a much bigger control over the process because he dominates the time and space when using a visual browser. In the second case, it is the computer that determines the rhythm of the voice browser, by phone (or any equivalent system, such as VoIP) and controls the time/space of the process.

Besides that, in the case of visual browsers, the simultaneous windows and processing allow the multiplication of the user identities in the cyberspace through the simultaneous persistence of several windows and processing. In the case of oral processes, even if we opt to follow a link to an option and then come back to the previous context, there is no way to keep both oral contexts simultaneously. In a moment we are in one context, in the other moment we are in another. Different oral contexts cannot persist simultaneously due their dependence of the time – the voice transience. Therefore, although voice interfaces allow the hypermidiatic access to a bigger context, they limit some aspects of the interaction that are usually possible through visual interfaces.

Still due to the transience, invisibility and asymmetry inherent to oral processes in voice interfaces, the limitation in the information processed, and that determines the possibilities of interactions, also differ from the traditional computational voice interfaces. For example, the search for a keyword is perfectly possible in a voice interface as much as in a visual one. However, what is the limit of information that we can analyze via an oral result? According to Kerchhove (2003: 20), the difficulty of closing the process is bigger in the oral case. Therefore, voice interfaces make it harder to process large amount of information due the peculiarities of orality.

In this context, the balance between control/pleasure and frustration of the user stays in a fragile zone. According to Murray (2003: 127), "When the things we do bring tangible results, we experience the second pleasure typical of electronic environments – the sense of agency. Agency is the rewarding capability of realizing significant actions and seeing the results of our decisions and choices". When the volume of information and rhythm of voice interfaces allow the closing and control of the process by the user, the agency pleasure really happens. However, a slight deviation that may prevent the control by the user in the agency process and its consequent pleasure can cause frustration and even abandonment of the process. The challenges are big, but no more than the possibilities that rise in the horizon of voice interfaces.

The experimentation of those limits and the combination of possibilities bring additional options to applications that can be explored to deliver richer user interfaces, improving the user experience and increasing the accessibility level.

2.4 Art as tool for experimenting voice interfaces

In this context, in 2004, it was created the *Voice Mosaic* – a web-art work that allows voice interactions on the web through the telephone, causing border dissolution between Internet and telephone. As said once by Hendrik Willem Van Loon (1937), "*The arts are an even better barometer of what is happening in our world than the stock market or the debates in congress*." and we believe that artworks help people to understand and experience the new emergent techno-social world that surround us, where convergence and hybridization have become ubiquitous and easy, and "to talk to computers or the web" is going to become common.

Since the technologies used in *Voice Mosaic* can be used in other kinds of voice applications on the web, improving accessibility and digital inclusion, we will present next the work and its main aspects, regarding either the art concept or the technological implications. This artwork received several awards and was also presented at SIGGRAPH Art Gallery 2006, in Boston, MA (USA).

3. Voice mosaic

The *Voice Mosaic* (figure 2) is a web-art application that combines speech and image, building a visual mosaic on the web with the chosen colors and recorded voices of people who interact with it from any place in the globe. The voice interface, developed with open-standards in speech synthesis and voice recognition technologies (VoiceXML), works through phone calls from any telephone – mobile or not. To participate in English, call in US: (800) 289.5570 or (407) 386-2174 / PIN number: 9991421055. The mosaic is accessed on the web at www.voicemosaic.com.br.

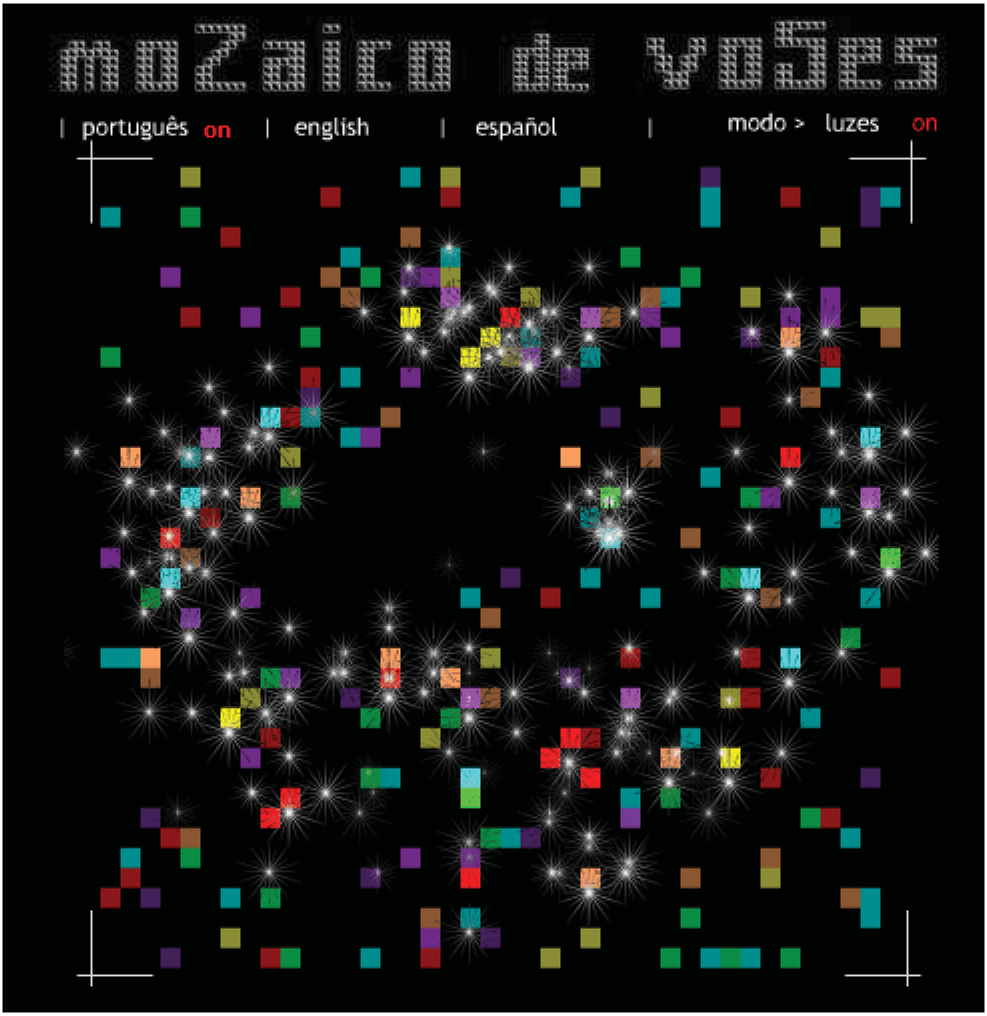


Fig. 2. Screeshot of the artwork Voice Mosaic showing the tiles

The application was developed in 2004, in three languages – Portuguese, English and Spanish - in order to encourage global participation. The phone calls form the mosaic on the web, and it happens spontaneously, therefore the mosaic changes as time goes on and its ongoing aesthetics and final result are unpredictable.

In this context, the work causes time-space collapse, and maps in one screen the participations that comes from several different geographical places, in different languages, and different times. Furthermore, using the search field, one can easily locate his/her participation by searching his/her own phone number. Also, one can locate all tiles in the mosaic within the same telephone area, which means to map geographical participations in the visual work.

The work puts together several dualities that do not oppose each other, but complete each other: speech / image, simple / complex, old / new, low-tech / high-tech, time / space, individual / community, passive / active, expected / uncertain, among others, in order to cause reflection and awareness about talking to the web, media convergence and hybridization between the telephone and the web.

3.1 Interfaces and technology

The work has two interfaces (see figure 1) – the voice interface accessed by phone and the web interface. As the web interface uses common and well known technologies – html, data base and Flash --, we will focus here on the voice interface, which is the core of the system.

The voice interface works via phone (mobile or not) interacting with the web. It is developed with VoiceXML, a structured language that offers support to build dialogs. When accessed by phone, the interface uses a Voice Gateway which allows voice recognition and speech synthesis during the conversation.

During the interaction by phone the person talks to the interface, choosing a color and recording a free speech message.

There are seven options available for choosing the color. This number, seven, is due the limit of information that a person can hold in the short-term memory. As mentioned previously in this text, according to Miller (1956) and explained in Zakia (1997), "There is a limit to the amount of unrelated information a person can hold in short-term memory (STM), from five to nine items, averaging seven. (…) Since we are limited in the amount of information we can retain correctly in STM, one should be cautious with the amount of information included in a multimedia program if it is going to have some memorable impact".

The free speech message is limited to 15 seconds because of the web interface where it will be listened – recorded files longer than 15 sec. would generate WAV files larger than 100kb, which is the maximum file size to allow a comfortable user experience while clicking and listening to the mosaic tiles without waiting too long to start playing.

The voice interface was designed using both pre-recorded human voice (in the welcome message) and synthesized text-to-speech voices to instruct the user, in order to cause the experimentation of the differences and similarities between them. Also, it is used touch tone and speech tone interactions in order to put side by side voice recognition (human-like feature) and touch recognition (machine-like feature) intending to cause reflection about the two ways of interacting by phone – talking and dialing.

In order to allow data visualization either by tracking or by locating the interactions in the visual mosaic, the voice interface records the Caller ID phone number. Due that we can know where the interactions come from in the globe and also locate all the interactions from within a specific area code. This reveals the space collapse in the mosaic on the web.

The phone calls, through the voice interface, are the way the data (and people) enter the *Voice Mosaic* on the web. No data enters the work via its web interface, which is used only for purposes of data visualization, interpretation and reflection.

4. Conclusion

The web and telephone have been the realm for the state of the art in voice technologies.

Voice Mosaic is on the web, and it has received voice participation for more than two years now, summing up about 800 tiles. Although we could realize that people do not know much about the technology they are experiencing in the work, they use it easily and get excited about "talking to the web" and becoming immediately a permanent tile there. We also realized that technical people (IT, engineers, etc.) were more resistant to first experiment with the work than lay people. The kind of messages people create is also interesting – they range from recorded music and people singing to love declarations and creative use of the voice.

The same kind of VoiceXML based voice interface created for the artwork Voice Mosaic can be used for any kind of application on the web, allowing people to "talk" to the web instead of only seeing it. This ability of dialoging with the web provides a better experience for users with visual disabilities while navigating online.

From now on we think that it will be possible to provide wider and deeper experimentation with voice interfaces due to the available technologies integrating the web and telephone. We expect it will probably allow us all to break frontiers and go further in human accessibility and digital inclusion developments.

5. References

Cohen, Michael; Giangola, James; Balogh, Jennifer. Voice User Interface Design. Boston, Addison-Wesley. (2004).

Christian, Kevin; Kules, Bill; Youssef; Adel. A Comparison of Voice Controlled and Mouse Controlled Web Browsing. (2000). Available at [http://otal.umd.edu/SHORE2000/voicebrowse/]. Access on 27.sept.2005.

Farber, D. 2014: Magic Software, Free Hardware, In ZDNet.com. (2004). Available at [http://techupdate.zdnet.com/techupdate/stories/main/Gates_gives_magical_software_tour.html].

Johnson, Steven. Cultura da Interface: como o computador transforma nossa maneira de criar e comunicar (2001). Tradução: Maria Luiza X. de A. Borges. Rio de Janeiro, J. Zahar.

Kerckhove, Derrick. A Arquitetura da Inteligência: Interfaces do corpo, da mente e do mundo, In: DOMINGUES, Diana (org.). *Arte e Vida no Século XXI*. São Paulo, Editora Unesp, p. 15-26.(2003).

Loon, H.W.V. *The Arts,* (1937).

Miller, G. *The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information*, In: Psychological Review, 63, 81-97. (1956).

Murray, Janet H. Hamlet no Holodeck: o futuro da narrativa no ciberespaço. (2003). Tradução Elissa Khoury Daher e Marcelo Fernandez Cuzziol. São Paulo, Itaú Cultural.

Nass, C. & Brave, S. Wired for Speech: How Voice Activates and Advances the Human-Computer Relationship. The MIT Press. (2005).

Nielsen, J. Voice Interfaces: Assessing the Potential. Available in [http://www.useit.com/alertbox/20030127.html]. (2003). Access on aug.30th.2005.

Palazzo, Luiz. Sistemas de Hipermídia Adaptativa: Fundamentos, Tecnologias e Aplicações (2002). Available at [http://gpia.ucpel.tche.br/~lpalazzo/sha/]. Access on 23.aug.2004.

Perkowitz, S. *Digital People: From Bionic Humans to Androids.* Washington: Joseph Henry Press, (2004).

Pinker, Steven. (2002). O Instinto da Linguagem – Como a mente cria a linguagem. São Paulo, Martins Fontes.(2002).

Santaella, Lucia. Matrizes da Linguagem e Pensamento – Sonora, Visual, Verbal. São Paulo, Iluminuras. (2001).

Wilson, S. Information Arts. Boston, MIT Press. (2002).

Wilson, S. Intersections of Art, Technology, Science & Culture – Links. Available at [http://userwww.sfsu.edu/~infoarts/links/wilson.artlinks2.html]. (2005). Access on jan.10th.2006.

Zakia, R. *Perception and Imaging*. Focal Press, (1997).